

AD-A100 736

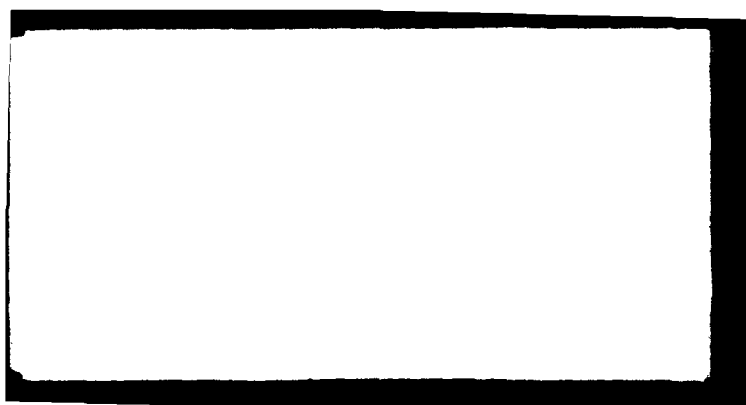
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/G 9/2
APPLICATION OF DISCRIMINANT ANALYSIS TECHNIQUES TO 6-STRESS ANA--ETC(U)
DEC 80 N E LEVINE
AFIT/GCS/EE/80D-11

UNCLASSIFIED

NL

1 OF 1
AD-A
100736

END
DATE
FILMED
7-81
DTIC

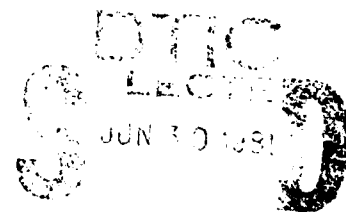


AFIT/GCS/EE/80D-11

APPLICATION OF DISCRIMINANT ANALYSIS TECHNIQUES
TO G-STRESS ANALYSIS
IN COMPUTER VOICE DECODING

AFIT/GCS/EE/80D-11

Nadine E. Levine
Capt USAF



A

Approved for public release; distribution unlimited.

(6) APPLICATION OF DISCRIMINANT ANALYSIS TECHNIQUES
TO G-STRESS ANALYSIS
IN COMPUTER VOICE DECODING.

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

(10) Nadine E. /Levine

Capt USAF

Graduate Computer Systems

(11) December 1980

Approved for public release; distribution unlimited.

012225

(12) 75

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Preface

This work has been motivated by the research of Dr. Matthew Kabrisky, Professor of Electrical Engineering at the Air Force Institute of Technology. It is an attempt to identify the effects of high gravity on speech.

I wish to extend a special thanks to Dr. Kabrisky and Lt Col McNichols, Professor of Operations Research at the Air Force Institute of Technology, for their advice, guidance and patience during the initial stages of this thesis which were more of a learning experience than a research project. I also wish to thank Mr. Jack D. Capehart of ASD/AD for his help in the preliminary processing of the analog speech data, and Mr. Don McKecknie of the Air Force Medical Research Lab for supplying high gravity speech data. Also Capt William Nelson, Mr. Denzel Henderson and Lt Carl Steven Lizza for their help with the day-to-day problems encountered when working on a computer.

Finally, I wish to thank Lt Pete Raeth and Maj Lester Holmes for being my friends, and supplying moral support and encouragement when I needed it most. And of course, the most important thanks of all, to my father and mother, who made me what I am.

Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1
Background	1
Objective	2
Basic Approach	2
Scope	4
II. Data Acquisition	6
Raw Data Acquisition	6
Analog-to-Digital Conversion and FFT	7
Spectrogram and Data Selection	7
III. Data Base Creation	10
Data Files and Structure	10
IV. Discriminant Analysis	13
Discriminant Analysis Capabilities	13
Tests Selected	15
Test Specifications (Data Base 1)	16
V. Discriminant Analysis Initial Results (Data Base 1)	19
VI. Data Base 1 Conclusions	22
VII. Data Base 2	24
Data Structure	24
VIII. Discriminant Analysis Processing (Data Base 2)	25
Tests Selected and Specifications	25
IX. Discriminant Analysis Results (Data Base 2)	27
X. Data Base 2 Conclusions	32

Contents

	Page
XI. Recommendations	33
Bibliography	35
Appendix A: Computer Programs	36
Vita	64

List of Figures

Figure		Page
1	Linearization of Spectrographic Data	3
2	Sample Spectrograph	8
3a	One Dimensional Data	14
3b	Two Dimensional Data	14
3c	Three Dimensional Data	14

List of Tables

Table		Page
I	TALK16 and TALK19 Frequency Channels	11
II	Summary of Results	20
III	Summary of Results - Data Set 2	28

Abstract

↙

The Statistical Package for the Social Sciences (SPSS) Discriminant Analysis Routines were applied to speech data obtained at various gravity levels. The data were created on the Air Force Medical Research Lab's Centrifuge at Wright-Patterson Air Force Base. They were then digitally sampled and Fourier transformed to 64 frequency bands which were converted to two separate files; one the 16 frequencies used by Guyote and Sisson; one the 19 frequencies used by Threshold Technology, Inc. These files were then further processed: summing the energy across time slices, summing across frequency bands, and averaging across frequency bands. Statistical results showed that G-stress seemed to be moving energy across time slices, more than shifting across frequencies.

↘

APPLICATION OF DISCRIMINANT ANALYSIS TECHNIQUES

TO G-STRESS ANALYSIS

IN COMPUTER VOICE DECODING

I. Introduction

Background

Despite the best efforts of psychologists and aviators to lay out control panels in an easy-to-read display, the amount of visual information available to a pilot is overwhelming, particularly in a one person fighter aircraft where the pilot may need to do more than keep the plane level. In addition, the number of buttons, knobs, levers and switches necessary to control an aircraft keeps the plane from reacting at any speed greater than the manual dexterity of the pilot. With the introduction of a new channel of information transfer, i.e., speech, the pilot could both process more information and control more aircraft functions at the same time, thereby increasing the plane's overall reaction capabilities.

Although the field of speech recognition has advanced to the point where isolated words can be recognized with a 99 percent accuracy under carefully controlled conditions (Ref 2), the aircraft environment has many speech distortion factors, among them: G-forces, oxygen mask, noise, vibration, and pilot stress and fatigue.

Threshold Technology has a voice recognition system with 99.32 percent recognition rate for normal voiced words and a better than 97 percent accuracy rate for commands spoken through a standard A.F. oxygen mask and microphone (Ref 10). However, there are no systems yet

capable of recognizing voiced speech distorted by high G-forces with any reliable degree of accuracy.

Research done at Rome Air Development Center from 5 January 1976 to 4 January 1977 using data gathered in the Brooks AFB centrifuge verified the existence of the G-stress problem, but failed to find any pattern in the distortion (Ref 4).

Objective

The objective of this project was to apply the Statistical Package for the Social Sciences (SPSS) Discriminant Analysis Techniques (Ref 6) to data obtained under G-stress, and, using the results, to form a pattern of the G-stress distortion.

Basic Approach

To use the SPSS package on speech data an immediate problem had to be surmounted. Speech data are often processed in spectrographic form, three dimensional arrays which can be pictured as 2-D graphs of frequency versus time with intensity as the third dimension. SPSS Discriminant Analysis is designed to work on linear vectors. Cutting the spectrographs into series of vectors would not solve the problem because the number of variables would be too great for the number of samples of data. SPSS would simply truncate the data arbitrarily.

To solve this problem, the spectrographic data were "linearized" in the following way (see Figure 1). The amplitudes of all frequency components within a time slice were added together forming a vector of Sum of Time Slice (SUMT). Likewise the amplitude of all the time slices for a single frequency were summed yielding a Sum of Frequency

Vector (FSUM). Since every word had its own length, the FSUM vectors were partially influenced by time, and so a third vector was created, the Average of Frequency Vector (FAVE), by dividing each FSUM vector by the number of time slices over which it was summed. Each of these vector types became the basic record for a file. The Discriminant Analysis Tests were run against each of these files and the results compared to find the G-stress effect.

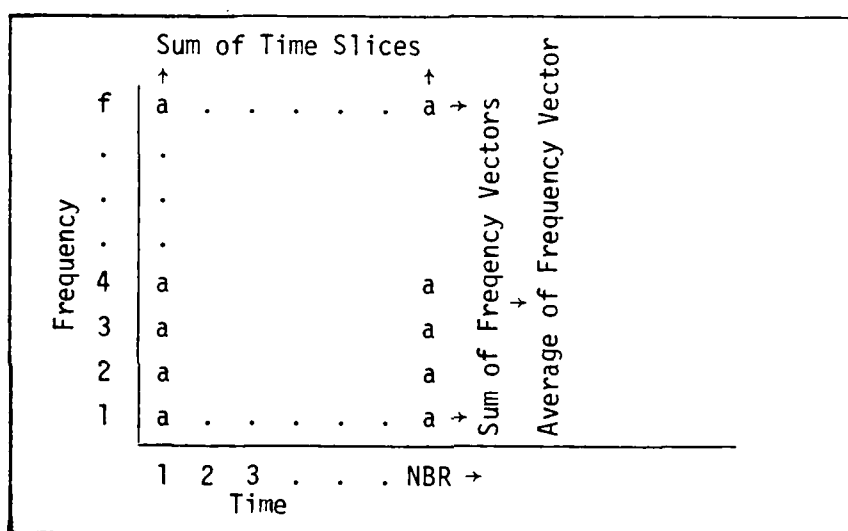


Figure 1. Linearization of Spectrographic Data

The discarding of information in each of these vector files causes a type of information "hiding" which is the key to the results of this thesis. It is best explained by imagining that all of the information contained in the speech data could be classified into say ten distinct components. Now suppose ten files are created, each one missing one of those components of information, and Discriminant Analysis tests are run for each of the files. If the results of those tests showed that one file had no distinctions between the data at different gravity levels it would clearly indicate that the G-stress

was affecting only that one component of information that was missing. In the data files used in this thesis, the FSUM files are missing information about the relative timing of energy amplitudes. The FAVE files are missing this and the total energy/length of word information. The SUMT files are missing information on the relative frequency amplitude ratios.

Scope

The scope of this project was limited to seven single word commands spoken by two volunteers at three basic gravity levels. The seven words used in this project were:

WAITING, BENCH, FATHERS, MY, OUT, AT, WAS

The gravity levels were 1G, 1.5G and 5G for the first speaker and 1G, 1.5G and 4G for the second speaker.

The data were broken into two data sets. The first included only the first three words (WAITING, BENCH, FATHERS) spoken by the first speaker (1G, 1.5G, 5G). The Discriminant Analysis results obtained from this limited data suggested the pattern of energy change over time. The second data set was used to further test this theory. It included the first three words spoken by the second speaker, the last four words spoken by the first speaker, and all seven words spoken by a third speaker not wearing an oxygen mask. The data sets seem small, however, each one completely filled a 7-track, 2400 ft, 800 bytes per inch tape when digitized.

All data input from the first two speakers were obtained using the standard A.F. oxygen mask and microphone. Although the mask itself creates a speech distortion, the presence of the mask at all

three gravity levels assures that only the G-stress distortion on the entire speaker-mask system will be measured since any distortion caused by the effect of high gravity on the mask or on the speaker, will show up in the data for analysis. The 1G, non-masked speech data in the second data set were used for analysis control.

II. Data Acquisition

Data acquisition was performed in the same basic manner as described by Guyote and Sisson (Ref 1).

Raw Data Acquisition

The original data tape used in this research was supplied by Mr. Don McKecknie of the Air Force Medical Research Lab of Wright-Patterson Air Force Base. Two volunteers were recorded in the centrifuge on two different channels of the same tape. The quality of this tape was somewhat less than ideal because of a high degree of breath noise in the oxygen mask microphone.

Each of the speakers recorded the same set of twelve words:

WAITING, BENCH, FATHERS, LAWN, SHE,
TOOK, SHOE, JOE, MY, OUT, AT, WAS

They were recorded first at 1G, with the centrifuge standing still, then at 1.5G, considered a baseline level for the centrifuge. Each volunteer was then given control for the high gravity run. The first speaker took the centrifuge to 5G and was able to repeat two samples of each word during his 30 second run. The second speaker recorded three samples each at 4G.

This original tape was then edited with the use of an entertainment grade tape cassette player and re-recorded on an unused portion of the original 1/4 inch tape. The use of this equipment was justified since the end result sounded the same to the human ear, indicating that all the information necessary for word recognition remained. In fact, the words used in the limited data space available were determined by their clarity to the human ear. For ease of

recognition of critical points in the tape, a short harmonica tone was recorded between speakers and gravity levels. These notes showed up very clearly on the spectrogram isolating data words from the surrounding noise.

Analog-to-Digital Conversion and FFT

The edited data were digitized by Mr. Jack Capehart of ASD/AD following the same procedures originated by Maj Neyman (Ref 5). These procedures resulted in an effective sampling rate of 10 kHz with the speech signal low pass filtered to 5 kHz.

The digitized speech was then Fast Fourier Transformed (FFT) using a window of 12.8 ms, or 128 samples per FFT. This resulted in a set of 64 discrete amplitude values, representing 64 frequency multiples of 78.125 Hz for each 12.8 ms time slice. Three minutes worth of this data could be stored on one tape.

Spectrogram and Data Selection

Spectrographic printout of the entire data tape was accomplished using a Fortran program provided by Capt William Nelson, AFIT/EN (see Appendix). This program was an improvement over the one used by Maj Neyman (Ref 5) in that it used only a single overprint to create the spectrographic light/dark effect. This program was also flexible enough to be used, with only minor variations, on the various data files created from the original tape.

The data representing the spoken words were then isolated from the surrounding noise. Beginning and end points of words were determined by visual study of the data tape spectrogram (see Figure 2)

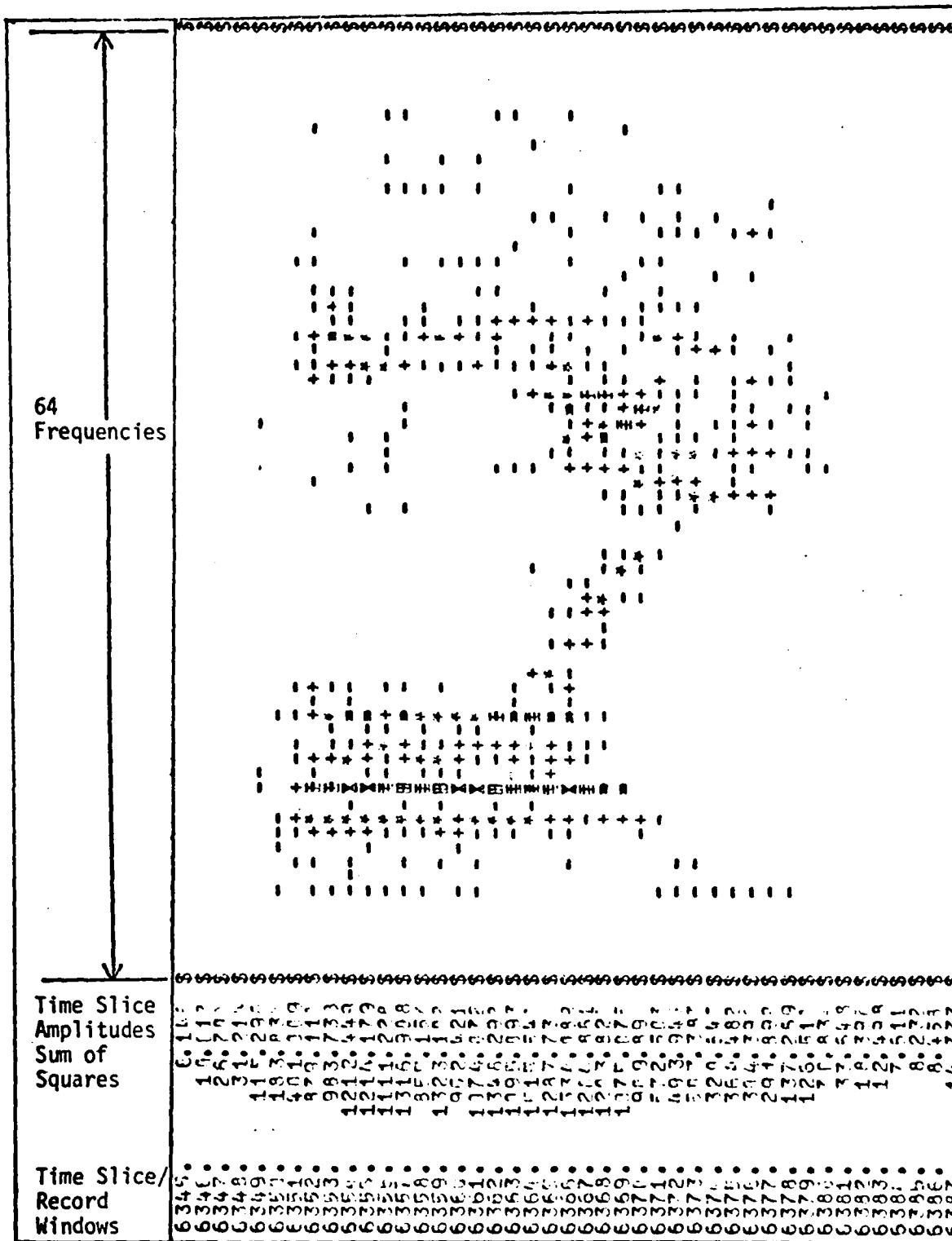


Figure 2. Sample Spectrograph
(Word is 'MY' at 1G with no mask, Data Set 2)

which included tape record numbers and sum of squares of the frequency amplitudes for each record. Precedence for this procedure can be found in Reference 7.

Program GETDATA (see Appendix) then created the first main data file (named TALKDATA), stored on disk for fast access. It did this by alternately bypassing groups of records on the tape representing noise, and copying groups of records from the tape which contained the spoken word data. (Each record on the tape contained 64 Fourier Transform values representing one 12.8 ms time slice.) Within the copying routine, a call to subroutine MAKEPF "standardized" the size of each word. For example, suppose one utterance of a word at a given gravity level lasted 28 time slices and another utterance lasted 33 time slices. The first word would have two time slices created by averaging time slices 10 and 11 and 20 and 21. The second would have time slices 9, 17 and 25 removed. Both words would then be exactly 30 time slices long, the template size representing that word and gravity.

III. Data Base Creation

Data Files and Structure

From the 64 channel file TALKDTA, two smaller files were created: TALK16 with 16 frequency channels, and TALK19 with 19 frequency channels (see Table I). These files were created both because 64 channels contained too many variables for the SPSS package to handle properly with the available number of data samples, and because the amplitudes of the higher frequencies were so small compared to the amplitudes of the lower frequencies, that they could not be effectively used as discriminating variables.

TALK16 was created by program CUTDOWN (see Appendix) using the same methods as Guyote and Sisson (Ref 1). It logarithmically increases the values of the higher frequency amplitudes while cutting down the number of frequency channels. This set of 16 frequency channels was selected for study in this project because of its common use in other speech recognition schemes.

TALK19 was created by program CONVERT (see Appendix). This set of 19 frequency channels was selected for study because Threshold Technology declared it to be the best set of input frequencies for its speech recognizer, which has been used successfully in speech recognition tests where the speakers were wearing oxygen masks (Ref 8).

The following procedures were used to create file TALK19:

1. The Threshold Technology Voice Control Demonstration System Operating and Maintenance Manual (Ref 9) displayed energy outputs for their filters at a given frequency. From this information, a curve representing the frequency response of each frequency could be plotted.

Table I

16 Channels (as per 1977 Thesis by M. F. Guyote and P. L. Sisson)	19 Channels (used by Threshold Technology Voice Control Demonstration System Equipments Manual Schematic & Parts List)
1. 78.125 Hertz	1. 260 Hertz
2. 156.250 Hertz	2. 395 Hertz
3. 234.375 Hertz	3. 535 Hertz
4. 312.500 Hertz	4. 683 Hertz
5. 390.625 Hertz	5. 841 Hertz
6. 468.750 Hertz	6. 1011 Hertz
7. 585.940 Hertz	7. 1198 Hertz
8. 742.188 Hertz	8. 1405 Hertz
9. 898.440 Hertz	9. 1635 Hertz
10. 1132.810 Hertz	10. 1892 Hertz
11. 1445.310 Hertz	11. 2179 Hertz
12. 1793.380 Hertz	12. 2505 Hertz
13. 2226.560 Hertz	13. 2885 Hertz
14. 2812.500 Hertz	14. 3326 Hertz
15. 3554.690 Hertz	15. 3855 Hertz
16. 4453.125 Hertz	16. 4484 Hertz
	17. 5263 Hertz
	18. 6277 Hertz
	19. 7626 Hertz

2. Using standard numeric analysis curve fitting techniques, the relative amplitudes (from 0 to 1) of the basic 64 frequency channels could be calculated and summed.

3. The resulting 19 amplitudes represent what the Threshold Technology filters would have recorded given the same speech input represented by the 64 channel file.

Program SUMFREQ (see Appendix) was then used, once for each file TALK16 and TALK19, to create the final Data Base Sets. Word parameters were read in first (WORD, GRAVITY, SPEAKER, and WORD LENGTH) followed by the time/frequency amplitudes in their Number of Frequency x Number of Time Slices data arrays. These arrays were then summed and averaged as explained in the introduction (see Figure 1 again). The final resulting files were labeled FSUM16, FAVE16, SUMT16, FSUM19, FAVE19, SUMT19 for the first data set. The second data set consisted of FSUM216, FAVE216, SUMT216, FSUM219, FAVE219, SUMT219 (the "2" signifying second data set).

In the first data set, the first and third words were 30 time slices long. The second word was only 18 time slices long, but was zero filled to create 30 numbers for the SUMT vectors. In the second data set, the longest word was 60 time slices and so 60 became the template size. All other SUMT data words, which had shorter "standard" time slice duration, were fitted into the beginning of the 60 time slice template, with zeros used for fillers.

IV. Discriminant Analysis

Discriminant Analysis Capabilities

Discriminant analysis is a statistical method of determining distinctive group characteristics of known grouped data, and determining group membership of ungrouped data according to those characteristics. The way in which this is done is analogous to finding a view of all of the data which best separates the groups. Figure 3a represents one dimensional data such as age of employees, or salary levels. . . . In the case pictured, the dimension can easily discriminate between the two groups of data, those above zero and those below zero. Any new data can then be grouped as positive or negative.

Figure 3b represents data consisting of two variables, conveniently labeled x and y. Notice that if only x or y were used to discriminate among the groups there would be much group overlapping and group membership classification would be difficult. However, if x and y are used, the group memberships become clearly distinct.

Figure 3c represents data sets of three dimensions. In the case pictured, neither x nor y can help discriminate between the groups, but the z data variable alone can clearly define group membership.

Clearly, data sets can have any "n" number of variables, and may be split into any "g" number of groups. The Discriminant Analysis Program can determine which variable best distinguishes among the groups and how much distinction it makes. It can then bring in the next most useful variable for group classification, and so on until either all the variables are used, the number of variables remaining

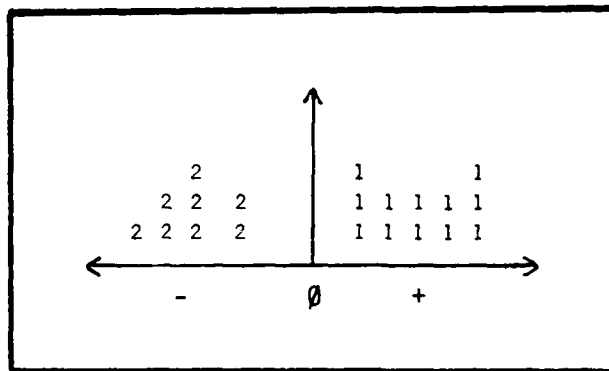


Figure 3a - One Dimensional Data

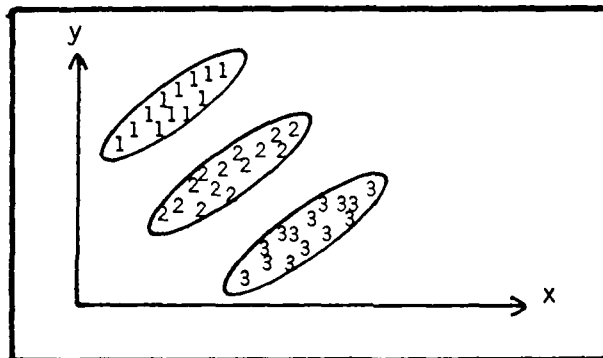


Figure 3b - Two Dimensional Data

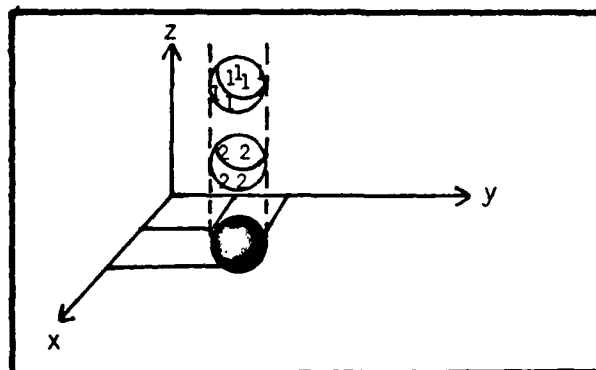


Figure 3c - Three Dimensional Data

add nothing to the discrimination, or the number of variables allowed has been reached.

There are many different ways in which the distinguishing variables can be chosen. In this thesis each test was run using two different methods: MAHAL and MAXMINF. This helped to determine if one method of selection was better than the other for any purpose, or whether the same variables would be consistently chosen as most discriminating. Method = MAHAL chooses variables that maximize the mahalanobis distance (similar to a Euclidian sum of squares) between group centroids. Method = MAXMINF chooses variables that maximize MINF, where F is a standard statistical two group F function and MINF is the minimum F value among all the inter-group F values. (A standard two group F function measures the ratio of the variance of two groups. Variance is a measure of the spreading or grouping of data on a graph such as in Figures 3a, 3b and 3c which show tightly grouped data.)

Tests Selected

The significance of tests was originally determined through trial and error. Initial efforts were designed to compare each word to itself at differing gravity levels, and then compare the resultant discriminant and classification functions to each other. However, no pattern was discerned from this method.

Next, the different words were compared against each other at each of the same gravity levels. The words at different gravity levels were then grouped according to the results of the classification functions derived from the determining set. From these tests it was

noted that the results for file FSUM16 were particularly good; good in that the classification functions derived from any one gravity level set of data were able to correctly identify the words of the other gravity levels.

The tests were then refined to limit the number of variables allowed for use in the classification functions. This was done both to increase the validity of the discrimination, and to determine which variables, if any, would be chosen consistently as discriminating characteristics.

Tests were also conducted to measure the distinguishability of each word if all gravity levels were used as a basis; and to separate each individual word-gravity combination as much as possible.

Test Specifications (Data Base 1)

Before the Group Discrimination Tests could be run, the data first had to be separated into groups. One obvious grouping simply used the variable IWORD to determine if a data record represented Word 1, 2 or 3. Another possible grouping would have been to divide the records (using IGRAV) into three groups based on gravity level alone.

Since it was possible to compute new functions for the purpose of group identification, the data records were also separated into nine groups representing each unique word/gravity pair (3 words x 3 gravity levels = 9 unique possible groups for Data Set 1). Two different naming functions were used to make reading of the SPSS results easier.

The function $IGRAVWORD = (IGRAV - 1) \times 3 + IWORD$ sets up nine groups which are semi-grouped by gravity level. They are:

Groups 1, 2, 3 = Words 1, 2 & 3 at 1G

Groups 4, 5, 6 = Words 1, 2 & 3 at 1.5G

Groups 7, 8, 9 = Words 1, 2 & 3 at 5G

The function $IWORDGRAV = (IWORD - 1) \times 3 + IGRAV$ sets up groups which are semi-grouped by word so that:

Groups 1, 2, 3 = Word 1 at 1, 1.5 and 5G

Groups 4, 5, 6 = Word 2 at 1, 1.5 and 5G

Groups 7, 8, 9 = Word 3 at 1, 1.5 and 5G

The test $IWORDGRAV (1,9)$ set the Discriminant Analysis Program to separate out each of the nine words/gravities as distinctly as possible. If the resultant groupings left the same word at different gravity levels close together, it would tend to indicate that the G-stress had little effect on that data set. Such a result should be accompanied by a low error rate on the $IWORD (1,3)$ test, which classifies the 3 word groups using only word classification for discrimination. Close grouping of the words at different gravity levels would lead to a low error rate, and would be due to a low G-stress effect.

The test $IGRAVWORD (1,3)$ compares $IGRAVWORD$ groups 1, 2 and 3 to each other and produces a set of classification functions which can discriminate among the three words at gravity level 1. SPSS then classifies all the other word data sets according to these same classification functions. These are called "ungrouped" data sets because they do not belong to groups 1, 2 or 3. Tests $IGRAVWORD (4,6)$

and IGRAVWORD (7,9) perform the same function as IGRAVWORD (1,3) using the gravity levels 1.5G and 5G respectively as the basis for classification.

V. Discriminant Analysis Initial Results (Data Base 1)

The results of the Discriminant Analysis Tests on Data Base 1 are summarized in Table II. The term VAR#### is a list of the variables used in the classification function. The term ERR refers to a word being incorrectly classified in any way. The term WW stands for Wrong Word and indicates that some word was misclassified as another. It is listed as a fraction in Part II of Table II indicating the number of words classified incorrectly/total number of ungrouped words classified.

Note that an error can occur that is not a wrong word if, for example, a word at 1G is classified as being the same word at 5G. The implication of this happening is that the discrimination program could not separate a word from itself at a different gravity level although it could easily tell which word it was. Alternately, a wrong word indicates that the gravity so changed a word that it was unrecognizable as itself. The effect of the G-stress on each of the Data Base 1 files could thus be measured as a function of the number of errors and the number of wrong words.

As can be seen, the results of test IWORDGRAV (1,9) indicate a distinctly higher error rate for the files SUMT16 and SUMT19 than for the other four files. The results of the Part II tests are not as marked. The difference in error rates between FAVE16, FSUM16, FSUM19 and SUMT16, SUMT19 follows the same pattern as the Part I tests, however, the error rate for FAVE19 is too high to fit the pattern.

Table II

Summary of Results

TEST		FIVE 16		FSUM 16		SUMT 16		FAVE 19		FSUM 19		SUMT 19	
WRODGRAV (1.9)	VAR: 3.5,9,10,13	0	VAR: 3.4,8,9,13	0	VAR: 3.6,12,18,26	3	VAR: 1,12,14,17	1	VAR: 1.9,10,17,19	0	VAR: 1.3,13,16,20	4	
	2 ERR	MM	0 ERR	MM	7 ERR	MM	1 ERR	MM	0 ERR	MM	10 ERR	MM	
WRODGRAV (1.9)	VAR: 3.4,5,9,10	2	VAR: 3.5,9,12,13	0	VAR: 3,12,18	6	VAR: 1,3,6,14,17	0	VAR: 2,3,9	2	VAR: 1,3,13,16,17	5	
	3 ERR	MM	1 ERR	MM	12 ERR	MM	0 ERR	MM	7 ERR	MM	9 ERR	MM	
WRODGRAV (1.3)	VAR: 7.8,10,12,13	0	VAR: 1.6,8,12,14	0	VAR: 3.16,19,23,25	0	VAR: 1.2,3,11,17	1	VAR: 1.3,5,11,19	0	VAR: 5.21,23,25,27	1	
	0 ERR	MM	0 ERR	MM	0 ERR	MM	1 ERR	MM	0 ERR	MM	1 ERR	MM	
WRODGRAV (1.3)	VAR: 7.8,10,12,13	0	VAR: 1.6,8,12,14	0	VAR: 3.16,19,23,28	0	VAR: 1.2,3,11,17	1	VAR: 1.3,5,11,19	0	VAR: 5.21,23,25,27	1	
	0 ERR	MM	0 ERR	MM	0 ERR	MM	1 ERR	MM	0 ERR	MM	1 ERR	MM	
TOTALS													
PART I	5 ERR	2 MM	1 ERR	0 MM	19 ERR	9 MM	3 ERR	3 MM	7 ERR	2 MM	21 ERR	11 MM	

PART I

Table 11

21

VI. Data Base 1 Conclusions

The high number of wrong words selected for files SUMT16 and SUMT19 in Part I indicates that the gravity word distinction affect is stronger in these two files than in the other four files. This is exactly the opposite of what would happen if gravity was only shifting energy from one frequency to another. Therefore, although the frequency shift is not ruled out, it seems that energy is shifting across time slices to a greater extent than any frequency shifting that might be present. And even though the number of errors which are not wrong word choices is also high (which would indicate a similarity across the gravity levels), the Wrong Word rate for SUMT16 and SUMT19 is 50 percent of all errors whereas for the three files FAVE16, FSUM16 and FSUM19 the Wrong Word rate is less than 31 percent.

The same conclusions can be reached from the error rate results of Part II. The low error rates for files FAVE16, FSUM16 and FSUM19 indicate that classifications of words grouped at any one gravity will hold for those words at different gravities, indicating small gravity distortion across the frequency bands. The high error rate of SUMT16 and SUMT19 indicate gravity distortion across time.

A possible explanation for this effect might be that at higher gravity levels, the pilot must force speech out in a sharp exhalation of breath. This might cause greater energy emphasis at the beginning of an utterance, or otherwise alter the timing of the components of the utterance.

File FAVE19 is an enigma. In Part I it has a wrong word ratio of 100 percent, and in Part II it has a high error rate, behaving just

like the files SUMT16 and SUMT19. But since its data is completely time independent (representing the average energy per time slice of each frequency) this behavior does not fit the pattern of the other five files, and challenges the conclusions based on that pattern.

It was hoped that the results of the second data set would be according to the first pattern. But if the FAVE219 data behaved as file FAVE19 did, then the theory of G-stress distortion would have to take this consistency into account and determine why this file behaved thus.

VII. Data Base 2

Data Structure

Data files FAVE216, FSUM216, SUMT216, FAVE219, FSUM219 and SUMT219 were created in a similar manner to Data Base 1. The main difference, physically, is that some of the word data in this data base extended for 60 time slices, which when summed within time slices led to a SUMT216 and SUMT219 record greater than 150 characters. Since SPSS cannot handle such large records, these two files had to be rewritten as two record data sets. It was also necessary to sort the second data set in order to more easily interpret the SPSS results. Neither of these file differences had any effect on the data.

The main difference between the two data sets, logically, is that Data Set II contains seven words, three speakers and five gravity levels (gravity level 5 is actually 1G with the speaker wearing no mask).

Following is a list of Data Base 2:

Word 1 = WAITING	Speaker 2 at G-levels 1, 2, 4 and Speaker 3 at G-level 5
Word 2 = BENCH	Speaker 2 at G-levels 1, 2, 4 and Speaker 3 at G-level 5
Word 3 = FATHERS	Speaker 2 at G-levels 1, 2, 4 and Speaker 3 at G-level 5
Word 4 = MY	Speaker 1 at G-levels 1, 2 and Speaker 3 at G-level 5
Word 5 = OUT	Speaker 1 at G-levels 1, 2, 3 and Speaker 3 at G-level 5
Word 6 = AT	Speaker 1 at G-levels 1, 2, 3 and Speaker 3 at G-level 5
Word 7 = WAS	Speaker 1 at G-levels 1, 2, 3 and Speaker 3 at G-level 5

VIII. Discriminant Analysis Processing (Data Base 2)

Tests Selected and Specifications

Function IGRAWORD = [(IGRAV - 1) * 7 + IWORD] sets up the 35 groups:

- Groups 1-7 = all seven words at Gravity Level 1 (1G)
- Groups 8-14 = all seven words at Gravity Level 2 (1.5G)
- Groups 15-21 = all seven words at Gravity Level 3 (5G)
- Groups 22-28 = all seven words at Gravity Level 4 (4G)
- Groups 29-35 = all seven words at Gravity Level 5 (1G - no mask)

Function IWORDGRAV = (IWORD - 1) * 5 + IGRAV sets up the 35 groups:

- Groups 1-5 = Word 1 at all Gravity Levels
- Groups 6-10 = Word 2 at all Gravity Levels
- Groups 11-15 = Word 3 at all Gravity Levels
- Groups 16-20 = Word 4 at all Gravity Levels
- Groups 21-25 = Word 5 at all Gravity Levels
- Groups 26-30 = Word 6 at all Gravity Levels
- Groups 31-35 = Word 7 at all Gravity Levels

Note that not every group actually has data in it. For instance, there is no data for IGRAWORD groups 25-28 because speaker 2 only spoke Words 1, 2 and 3 and no other speaker was recorded at 4G's. This does not invalidate the test results. Whenever a group has no data, SPSS simply leaves it out of the classification function.

The tests run on Data Base 2 were selected to parallel the Data Base 1 tests. Test IWORDGRAV (1,35) is the same as Data Set 1's

IWORDGRAV (1,9) in that it discriminates each word/gravity group from the other. The much higher error rate of this test in the second data base is not because of the greater number of words, but rather the fact that each word was spoken by two speakers; one with no face mask at 1G, and the other with a face mask at various G-levels. The high error rate indicates that these words together do not form a coherent group distinctly different from the other word groups.

In contrast, the tests IGRAVWORD (x,y) all compare all seven words spoken at the same gravity level (and the same speaker). This compares exactly to the Data Base 1 tests.

IX. Discriminant Analysis Results (Data Base 2)

Table III shows the results of SPSS testing on Data Base 2. As before, WW stands for wrong word, ERR for errors, and VAR for variables chosen, and the results are in fractions, number wrong/number available.

An extra level of complexity has been added to the Data Base 2 results table due to the number of speakers. That is why there are two error fractions for each test. Each is labeled, either same speaker error rate or other speaker error rate (at the same gravity level or different gravity level). The totals have also been separated out in this manner, and the number of errors in the grouped data have also been accounted for.

The results of test IWORDGRAV (1,35) (Part I) follow the same pattern as for Data Base 1. Both SUMT216 and SUMT219 have a WW/ERR ratio of 74 percent, while FSUM216 and FSUM219 have a WW/ERR ratio of 46 percent. FAVE216 and FAVE219 are 66 percent and 63 percent respectively.

Likewise, the results of Part II totals for the same speaker also follow the same basic pattern of Data Set I, with SUMT216 and SUMT219 having distinctly higher error rates than FSUM216 and FSUM219. However, the error rates for FAVE216 and FAVE219 are only marginally lower than the error rates for SUMT216 and SUMT219. It can also be seen that the error rate for a different speaker (with/without fave mask) is approximately twice the gravity difference error rate for the same speaker. And a change of gravity level at that point does not raise the error rates.

Table III

Summary of Results - Data Set 2

[IGRAYWORD = (IGRAY - 1) * 7 + IWORD]

[IWORDGRAV = (IWORD - 1) * 5 + IGRAV]

TEST	FAVE216			FSUMT216			SUMT216			FAVE219			FSUMT219			SUMT219		
	VAR: 1.2,7,11,13 22 ERR	15 WM	VAR: 1.9,13,14,16 13 ERR	6 WM	VAR: 1.7,14,19,23 29 ERR	19 WM	VAR: 2.4,7,17,18 15 ERR	8 WM	VAR: 1.2,3,5,6 13 ERR	6 WM	VAR: 1.6,11,15,23 29 ERR	22 WM						
IWORDGRAV (1,35)																		
WAXMINF (5)																		
IWORDGRAV (1,35)																		
MAHAL (5)																		
IWORD (1,7)																		
WAXMINF (5)																		
IWORD (1,7)																		
MAHAL (5)																		

Table III (continued)

TEST	FAVE216	FSUM216	SUMT216	FAVE219	FSUM219	SUMT219
GRAYWORD (1,7) MAXINF (5)	VAR: 6.8,12,15,16 27/63 All 2's 23/27 VAR: 6.7,8,11,14 39/63 All 2's 23/27 VAR: 8.9,10,13,14 20/63 25/27 VAR: 4.8,9,10,13 19/63 Grouped 23/27 VAR: 4 only (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 6.8,10,11,12 22/63 26/27 VAR: 6.8,10,11,13 20/63 26/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.10,21 (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 10,22,25 4 HW Grouped 27/63 23/27 VAR: 4,10,21,25,20 1 HW Grouped 26/63 19/27 VAR: 2,3,12 (F-Level Tolerance) 2 HW Grouped 10/24 8/12	VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,5,6,16,17 18/63 27/27 VAR: 1,5,6,16,17 11/63 27/27 VAR: 1 (By Removal) 2 HW Grouped 17/24 21/27 8/12	VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1 (only 3 steps by accident) 2 HW Grouped 17/24 21/27 8/12	VAR: 5,10,11,14,27 2 HW Grouped 34/63 24/27 VAR: 5,10,20,30,39 27/63 19/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 3,7,19 (F-Level Tolerance) 2 HW Grouped 14/24 5/12
GRAYWORD (1,7) MAXINF (5)	VAR: 6.8,12,15,16 27/63 All 2's 23/27 VAR: 6.7,8,11,14 39/63 All 2's 23/27 VAR: 8.9,10,13,14 20/63 25/27 VAR: 4.8,9,10,13 19/63 Grouped 23/27 VAR: 4 only (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 6.8,10,11,12 22/63 26/27 VAR: 6.8,10,11,13 20/63 26/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.10,21 (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 10,22,25 4 HW Grouped 27/63 23/27 VAR: 4,10,21,25,20 1 HW Grouped 26/63 19/27 VAR: 2,3,12 (F-Level Tolerance) 2 HW Grouped 10/24 8/12	VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,5,6,16,17 18/63 27/27 VAR: 1,5,6,16,17 11/63 27/27 VAR: 1 (By Removal) 2 HW Grouped 17/24 21/27 8/12	VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1 (only 3 steps by accident) 2 HW Grouped 17/24 21/27 8/12	VAR: 5,10,11,14,27 2 HW Grouped 34/63 24/27 VAR: 5,10,20,30,39 27/63 19/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 3,7,19 (F-Level Tolerance) 2 HW Grouped 14/24 5/12
GRAYWORD (8,14) MAXINF (5)	VAR: 6.8,12,15,16 27/63 All 2's 23/27 VAR: 6.7,8,11,14 39/63 All 2's 23/27 VAR: 8.9,10,13,14 20/63 25/27 VAR: 4.8,9,10,13 19/63 Grouped 23/27 VAR: 4 only (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 6.8,10,11,12 22/63 26/27 VAR: 6.8,10,11,13 20/63 26/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.10,21 (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 10,22,25 4 HW Grouped 27/63 23/27 VAR: 4,10,21,25,20 1 HW Grouped 26/63 19/27 VAR: 2,3,12 (F-Level Tolerance) 2 HW Grouped 10/24 8/12	VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,5,6,16,17 18/63 27/27 VAR: 1,5,6,16,17 11/63 27/27 VAR: 1 (By Removal) 2 HW Grouped 17/24 21/27 8/12	VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1 (only 3 steps by accident) 2 HW Grouped 17/24 21/27 8/12	VAR: 5,10,11,14,27 2 HW Grouped 34/63 24/27 VAR: 5,10,20,30,39 27/63 19/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 3,7,19 (F-Level Tolerance) 2 HW Grouped 14/24 5/12
GRAYWORD (19,21) MAXINF (5)	VAR: 6.8,12,15,16 27/63 All 2's 23/27 VAR: 6.7,8,11,14 39/63 All 2's 23/27 VAR: 8.9,10,13,14 20/63 25/27 VAR: 4.8,9,10,13 19/63 Grouped 23/27 VAR: 4 only (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 6.8,10,11,12 22/63 26/27 VAR: 6.8,10,11,13 20/63 26/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.4,6,10,16 23/63 27/27 VAR: 1.10,21 (F-Level Tolerance) 1 HW Grouped 13/24 All 2's 10/12	VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 5,10,11,14,18 3 HW Grouped 21/27 VAR: 10,22,25 4 HW Grouped 27/63 23/27 VAR: 4,10,21,25,20 1 HW Grouped 26/63 19/27 VAR: 2,3,12 (F-Level Tolerance) 2 HW Grouped 10/24 8/12	VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,2,11,14,16 21/63 All 2's 23/27 VAR: 1,5,6,16,17 18/63 27/27 VAR: 1,5,6,16,17 11/63 27/27 VAR: 1 (By Removal) 2 HW Grouped 17/24 21/27 8/12	VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 2,3,10,16,19 19/63 All 2's 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1,4,5,11,19 1 HW Grouped 13/63 23/27 VAR: 1 (only 3 steps by accident) 2 HW Grouped 17/24 21/27 8/12	VAR: 5,10,11,14,27 2 HW Grouped 34/63 24/27 VAR: 5,10,20,30,39 27/63 19/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 5,11,23,24,25 1 HW Grouped 25/63 21/27 VAR: 3,7,19 (F-Level Tolerance) 2 HW Grouped 14/24 5/12

29
62

Table III (continued)

TEST	FAVE216	FSUM216	SUMT216	FAVE219	FSUM219	SUMT219
TGRAVWORD (19,21) MAHAL (5)	VAR: 4(only)(F-Level Tolerance) Grouped 13/24 WW All 21's 8/12 WW VAR: 4,5,9,10,16	VAR: 1,10,12(F-Level Tolerance) Same Spkr WW 6/24 4/12 VAR: 1,8,16	VAR: 2,3,12(F-Level Tolerance) Same Spkr WW 10/24 8/12 VAR: 13,19,21,25,27	VAR: 1 (By Removal) Grouped 17/24 WW All 21's 8/12 WW VAR: 1,2,6,9(F-Level Tolerance) Same Spkr WW 17/32 7/11 VAR: 1,2,6,9	VAR: 1,3 (Removed) Grouped 15/24 WW All 21's 8/12 WW VAR: 1,3 (F-Level Tolerance) Same Spkr WW 6/32 All 21's 7/11 WW VAR: 1,3 (F-Level Tolerance) Same Spkr WW 6/32 All 21's 7/11 WW VAR: 4,5,11,13,18	VAR: 3,7,19 (F-Level Tolerance) Same Spkr WW 14/24 5/12 VAR: 4,7,8,21,26
TGRAVWORD (22,24) MAXMINF (5)	VAR: 4,5,9,10,16 17/32 4/11 VAR: 4,5,9,10,16	VAR: 1,9,12,14,16 Same Spkr WW 9/32 7/11 VAR: 1,9,12,14,16	VAR: 13,19,21,25,27 Same Spkr WW 15/32 5/11 VAR: 13,19,21,25,27	VAR: 1,2,6,9 Same Spkr WW 17/32 7/11 VAR: 1,2,6,9	VAR: 1,3 (F-Level Tolerance) Same Spkr WW 6/32 All 21's 7/11 WW VAR: 1,3 (F-Level Tolerance) Same Spkr WW 6/32 All 21's 7/11 WW VAR: 4,5,11,13,18	VAR: 4,7,8,21,26 Same Spkr WW 14/32 6/11 VAR: 4,7,8,21,26
TGRAVWORD (29,35) MAXMINF (5)	VAR: 5,6,11,12,14 15/24 WW 37/39 WW VAR: 5,6,11,12,14	VAR: 1,6,10,11,14 Same Spkr WW 17/24 31/39 VAR: 1,2,5,11,14	VAR: 13,18,22,33,34 Same Spkr WW 21/24 38/39 VAR: 13,18,22,33,34	VAR: 2,4,9,13,17 Same Spkr WW 18/24 36/39 VAR: 3,4,9,14,19	VAR: 4,5,11,13,18 Same Spkr WW 21/24 39/39 VAR: 4,5,11,13,18	VAR: 16,18,33 Same Spkr WW 19/24 38/39 VAR: 16,18,33
TGRAVWORD (29,35) MAHAL (5)	VAR: 5,6,11,12,14 15/24 WW 37/39 WW VAR: 5,6,11,12,14	VAR: 1,2,5,11,14 Same Spkr WW 15/24 32/63 VAR: 1,2,5,11,14	VAR: 13,18,22,33,34 Same Spkr WW 21/24 38/39 VAR: 13,18,22,33,34	VAR: 3,4,9,14,19 Same Spkr WW 19/24 34/39 VAR: 3,4,9,14,19	VAR: 4,5,11,13,18 Same Spkr WW 21/24 39/39 VAR: 4,5,11,13,18	VAR: 16,18,33 Same Spkr WW 19/24 38/39 VAR: 16,18,33

*No Errors (11/27)
for Same Spkr
(#3 - All 16 - No Mask)

**No Errors (11/27)
for Same Spkr
(#3 - All 16 - No Mask)

Table III (continued)

TOTALS - PART I													
TEST		FAVE216		FSUM216		SUMT216		FAVE219		FSUM219		SUMT219	
MAHLE		55 MW		33 MW		55 MW		57 MW		36 MW		58 MW	
MAHLE		53 MW		34 MW		58 MW		58 MW		37 MW		53 MW	
TOTALS - PART II													
MAHLE		1 ERR		0 ERR		7 ERR		2 ERR		3 ERR		3 ERR	
Same Speaker		77/182 = 42%		60/182 = 33%		90/182 = 49%		73/182 = 40%		55/182 = 30%		87/182 = 48%	
Same G. ERRS													
Totals		38/51 = 75%		43/51 = 84%		42/51 = 82%		41/51 = 80%		44/51 = 86%		43/51 = 84%	
Other Speaker													
Same G.													
Other Speaker		74/89 = 83%		69/89 = 78%		74/89 = 83%		78/89 = 88%		77/89 = 87%		70/89 = 79%	
Other G.													
MAHLE		2 ERR		0 ERR		4 ERR		2 ERR		2 ERR		1 ERR	
Same Speaker		87/102 = 48%		50/102 = 32%		89/102 = 49%		73/102 = 40%		53/102 = 29%		80/182 = 44%	
Same G. ERRS													
Totals		38/51 = 75%		41/51 = 80%		42/51 = 82%		42/51 = 82%		44/51 = 86%		38/51 = 75%	
Other Speaker													
Same G.													
Other Speaker		72/89 = 81%		70/89 = 79%		70/89 = 79%		76/89 = 85%		77/89 = 87%		70/89 = 79%	
Other G.													
TOTALS		3 ERR		0 ERR		11 ERR		4 ERR		5 ERR		4 ERR	
Same Speaker		164/364 = 45%		118/364 = 32%		179/364 = 49%		146/364 = 40%		108/364 = 30%		167/364 = 46%	
Same G. ERRS													
Totals		76/102 = 75%		84/102 = 82%		84/102 = 82%		83/102 = 81%		88/102 = 86%		81/102 = 79%	
Other Speaker													
Same G.													
Other Speaker		146/178 = 82%		139/178 = 78%		144/178 = 81%		154/178 = 87%		154/178 = 87%		140/178 = 79%	
Other G.													

31

X. Data Base 2 Conclusions

The results of Part I test I (discriminating each word/gravity pair from all the others) and the results of Part II totaled for the same speaker tend to support the theory that gravity is affecting energy across time slices more than across frequency bands.

However, the results are far from conclusive. If only the files FSUM216 and FSUM219 vs SUMT216 and SUMT219 were used in the analysis the results would be clear. However, the results of files FAVE216 and FAVE219 do not behave well according to the theory proposed. The possibility exists that the error rates for FSUM216 and FSUM219 are lower than the other files because they retain more of the information available in the original word spectrograms. Certainly files FAVE216 and FAVE219 have lost the information of the word length. However, it is unclear what information if any was lost to files SUMT216 and SUMT219 except the ability to discriminate among the frequencies. Since FSUM216 and FSUM219 cannot discriminate across time slices, it would seem that their relative error rates would indicate a valid pattern.

XI. Recommendations

Obviously, more data are needed before concrete conclusions can be drawn. The quality and content of that data should also be better controlled. For instance, for every speaker used in the experiment, the data base should contain speech at 1G - no mask, speech at 1G - with mask, centrifuge baseline with mask, and at least two other gravity level rides with mask. Also, these high gravity level rides should all be taken at the same gravity level for all riders.

A second recommendation is that more research be done on other ways to set up the data files. Specifically, what other components of the speech data can be isolated and extracted to create new files with new components of information hidden.

A third recommendation concerns the data results in Tables I, II and III. The variables used for each set of group classifications is listed for both Data Sets. There is some apparent heavy use of frequency 5 for FSUM216, FAVE216, FSUM16 and FAVE16 and a corresponding (though not equal) heavy use of frequency 2 in files FSUM219, FAVE219, FSUM19 and FAVE19. Channel 5 in the 16 frequency band = 390.623 Hz. Channel 2 in the 19 frequency band = 395 Hz. This shows that this frequency is a main discriminant between the various words. Further study might reveal other key frequencies. These frequencies might then form a basis for the information hiding studies just discussed.

Finally, some recommendations for work that could be performed on the data bases already available which was not performed simply because of a lack of time. Data Sets 1 and 2 could be merged into a third Data Set containing the first 3 words spoken at all available

gravity levels by all three speakers. This would make it possible to do preliminary studies on the effects of different speakers at the same gravity level with both wearing oxygen masks as compared to a different speaker without a face mask. (More data would be needed for more comprehensive studies of the separation of face mask, G-stress and speaker identity effects.) Also, all the available data words could be stretched or cut to a single template size (no filler) using the routines in program GETDATA (see Appendix). Not only could further tests be conducted on this data, but each word could be further normalized to have the same overall amount of energy by simply dividing each amplitude in each template by the energy sum total. This might eliminate so much information that one word could not be distinguished from another, or it might lead to very clearcut word independent G-stress effects. The only way to find out is to try it. [Some more elegant, but very time consuming computer routines have been employed by (at least) NEC and Lear-Siegler Corporation to handle this problem by a number of dynamic programming methods. This thesis would use a simpler approach since G-stress is the prime concern, not word recognition.]

Bibliography

1. Guyote, M. F. and P. L. Sisson. Computer Identification of Phonemes in Continuous Speech. AFIT Thesis, November 1979.
2. Lea, Wayne A. Critical Issues in Airborne Application of Speech Recognition. Speech Communications Research Laboratory, 1979.
3. McNichols, Charles W. Applied Multivariate Data Analysis (course notes). AFIT, 1979 (draft, unpublished).
4. Montague, Hill. Voice Control System for Airborne Environment. Scope Electronics, Inc., January 1977.
5. Neyman, Ralph W., Maj USAF. Computer Identification of Phonemes in Continuous Speech. Air Force Institute of Technology Thesis, June 1976.
6. Nie, Norman H., et al. SPSS (Statistical Package for the Social Sciences). New York: McGraw Hill Book Co., 1975.
7. Potter, Ralph K., et al. Visible Speech. New York: D. Van Nostrand Company, Inc., 1947.
8. Threshold Technology, Inc. Voice Control Demonstration System Equipment Manual Schematics and Parts List. Contract F33615-75-C-1075, New Jersey, March 1974.
9. Threshold Technology, Inc. Voice Control Demonstration System Operating and Maintenance Equipment Manual. Contract F33615-73-C-1095, March 1974.
10. Threshold Technology, Inc. Voice Control Demonstration System. Technical Report AFAL-TR-74-174, June 1974.

APPENDIX A

```

PROGRAM SUMFREQ (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4)

CCC
CCC THIS PROGRAM WILL CONVERT FILE TALK19 TO FSUM19
CCC WHICH WILL BE THE SUM OF EACH FREQUENCY ACROSS
CCC TIME. OUTPUT FORMAT WILL BE: IWORD,IGRAV,
CCC ISPKER, IWRDSZ, 19 FREQ SUMS.
CCC THIS WILL BE SUITABLE FOR INPUT TO SPSS.
CCC ALSO CREATED WILL BE SUMT19, WHICH IS THE
CCC SUM OF ALL FREQUENCIES WITHIN A
CCC TIME SLICE. FORMAT IS: IWORD,IGRAV,ISPKER,
CCC IWRDSZ, 60 SUMT'S.
CCC
CCC DIMENSION A(60,19), B(19), C(60), IC(60)

CCC
CCC IWORD = 1
CCC IGRAV = 1
CCC NBRTIM =60
CCC DO 190 ISPEC = 1,90

CCC
CCC CALL READA (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
CCC CALL SUMF (A,B,NBRTIM)
CCC CALL SUMT (A,IC,NBRTIM)
CCC WRITE (2,10) IWORD,IGRAV,ISPKER,IWRDSZ,(B(J), J = 1,19)
CCC WRITE (3,20) IWORD,IGRAV,ISPKER,IWRDSZ,(IC(I),I=1,60)
CCC WRITE (4,10) IWORD,IGRAV,ISPKER,IWRDSZ,((B(J)/IWRDSZ),J=1,19)
10 FORMAT (4I1,19F6.1)
20 FORMAT (4I1,60I4)

CCC
CCC IF (ISPEC.EQ.4) IGRAV = 2
CCC IF (ISPEC.EQ.8) IGRAV = 3
CCC IF (ISPEC.EQ.10) IGRAV = 1
CCC IF (ISPEC.EQ.10) IWORD = 2
CCC IF (ISPEC.EQ.10) NBRTIM = 18
CCC IF (ISPEC.EQ.14) IGRAV = 2
CCC IF (ISPEC.EQ.18) IGRAV = 3
CCC IF (ISPEC.EQ.20) IGRAV = 1
CCC IF (ISPEC.EQ.20) IWORD = 3
CCC IF (ISPEC.EQ.20) NBRTIM = 30
CCC IF (ISPEC.EQ.24) IGRAV = 2
CCC IF (ISPEC.EQ.28) IGRAV = 3
190 CONTINUE
STOP
END

CCC
CCC

```

```

SUBROUTINE READA (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
DIMENSION A(60,19)
READ (1,*) IWORD,IGRAV,ISPKER,IWRDSZ
DO 100 I = 1,NBRTIM
DO 100 J = 1,19
    READ (1,*) A(I,J)
100 CONTINUE
RETURN
END

CCC
CCC
SUBROUTINE SUMF (A,B,NBRTIM)
DIMENSION A(60,19), B(19)
DO 100 J = 1,19
    B(J) = 0.0
DO 100 I = 1,NBRTIM
    B(J) = B(J) + A(I,J)
100 CONTINUE
RETURN
END

CCC
CCC
SUBROUTINE SUMT (A,IC,NBRTIM)
DIMENSION A(60,19), C(60), IC(60)
DO 50 I = 1,60
    C(I) = 0.0
    IC(I) = 0
50 CONTINUE
DO 110 I = 1,NBRTIM
DO 100 J = 1,19
    C(I) = C(I) + A(I,J)
100 CONTINUE
    IC(I) = INT(C(I))
110 CONTINUE
RETURN
END

```



```

PROGRAM CUTDOWN (INPUT,OUTPUT,TAPE1,TAPE3)

CCC
CCC THIS PROGRAM READS IN TAPE1 (TALK3)
CCC WHICH CONTAINS 64 CHANNELS OF DATA AND
CCC COMPRESSES IT TO 16 CHANNELS ACCORDING
CCC TO GUYOTE AND SISSON 1977, CREATING
CCC TAPE3 (PERMFILE - SIXTEEN)
CCC
    DIMENSION A(60,64), B(60,16)
    NBRTIM = 60
    DO 100 I = 1,90
    CALL READIN (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
    CALL CONVERT (A,B,NBRTIM)
    CALL BOUT (B,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
100 CONTINUE
    STOP
    END

CCC
CCC
    SUBROUTINE READIN (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)

CCC
CCC THIS SUBROUTINE READS IN A AND HOLDS
CCC THE BEGINNING WORD/GRAVITY MARKER
CCC
    DIMENSION A(60,64)

CCC
    READ (1,*) IWORD,IGRAV,ISPKER,IWRDSZ
    DO 100 I = 1,NBRTIM
    DO 100 J = 1,64
        READ (1,*) A(I,J)
100 CONTINUE
    RETURN
    END

CCC
CCC
    SUBROUTINE BOUT (B,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)

CCC
CCC THIS SUBROUTINE WRITES OUT B TO TAPE3
CCC
    DIMENSION B(60,16)

CCC
    WRITE (3,*) IWORD,IGRAV,ISPKER,IWRDSZ
    DO 100 I = 1,NBRTIM
    DO 100 J = 1,16
        WRITE (3,*) B(I,J)
100 CONTINUE
    RETURN
    END

CCC
CCC

```

```

SUBROUTINE CONVERT(A,B,NBRTIM)
CCC
CCC  THIS SUBROUTINE CONVERTS A(60X64) TO
CCC  B(60X16) ACCORDING TO THE LOGRITHMIC
CCC  COMPRESSION SCALE USED BY GUYOTE AND
CCC  SISSON 1977 AFIT THESIS
CCC
      DIMENSION A(60,64), B(60,16)
      DO 900 I = 1,NBRTIM
        DO 100 J = 1,6
          B(I,J) = A(I,J)
100    CONTINUE
        J = 7
        DO 210 K = 7,9
          ADDJ = 0
          DO 200 N = 1,2
            ADDJ = ADDJ + A(I,J)
            J = J + 1
200    CONTINUE
          B(I,K) = ADDJ
210    CONTINUE
CCC
CCC    J = 13
CCC
        DO 310 K = 10,11
          ADDJ = 0
          DO 300 N = 1,4
            ADDJ = ADDJ + A(I,J)
            J = J + 1
300    CONTINUE
          B(I,K) = ADDJ
310    CONTINUE

```

```

CCC      B(I,12) = 0
          DO 400 J = 21,25
            B(I,12) = B(I,12) + A(I,J)
400      CONTINUE
CCC      B(I,13) = 0
          DO 500 J = 26,31
            B(I,13) = B(I,13) + A(I,J)
500      CONTINUE
CCC      B(I,14) = 0
          DO 600 J = 32,40
            B(I,14) = B(I,14) + A(I,J)
600      CONTINUE
CCC      B(I,15) = 0
          DO 700 J = 41,50
            B(I,15) = B(I,15) + A(I,J)
700      CONTINUE
CCC      B(I,16) = 0
          DO 800 J = 51,64
            B(I,16) = B(I,16) + A(I,J)
800      CONTINUE
900      CONTINUE
          RETURN
          END

```

```

PROGRAM CONVERT (INPUT,OUTPUT,TAPE2,TAPE3)

CCC
CCC THIS PROGRAM CONVERTS FROM THE 64 FREQUENCY
CCC FILE TALKDTA TO THE 19 CHANNELS USED BY
CCC THRESHOLD TECHNOLOGY AS INPUT TO THEIR
CCC SPEECH RECOGNIZER SYSTEM.
CCC
      DIMENSION A(60,64), B(60,19)
CCC
      DO 100 K = 1,90
      NBRTIM = 60
CCC
      CALL READIN (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
      CALL NEWFREQ (A,B,NBRTIM)
      CALL WRITEO (B,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
100 CONTINUE
      STOP
      END

CCC
CCC
      SUBROUTINE READIN (A,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
      DIMENSION A(60,64)
      READ (2,*) IWORD,IGRAV,ISPKER,IWRDSZ
      DO 100 I = 1,NBRTIM
      DO 100 J = 1,64
         READ (2,*) A(I,J)
100 CONTINUE
      RETURN
      END

CCC
CCC
      SUBROUTINE WRITEO (B,NBRTIM,IWORD,IGRAV,ISPKER,IWRDSZ)
      DIMENSION B(60,19)
      WRITE (3,*) IWORD,IGRAV,ISPKER,IWRDSZ
      DO 100 I = 1,NBRTIM
      DO 100 J = 1,19
         WRITE (3,*) B(I,J)
100 CONTINUE
      RETURN
      END

CCC
CCC

```

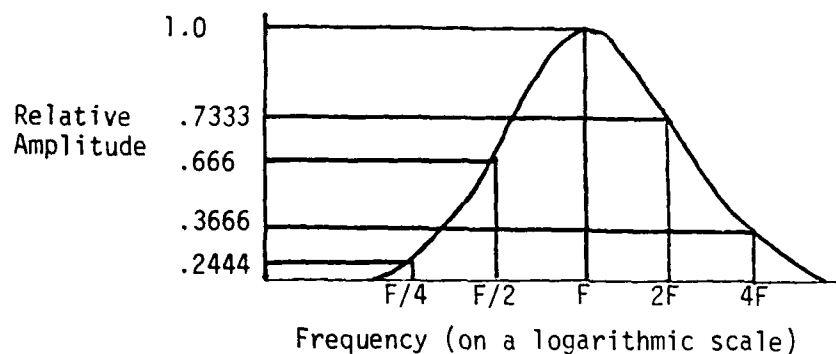
```

SUBROUTINE NEWFREQ (A,B,NBRTIM)
CCC
  DIMENSION A(60,64), B(60,19)
CCC
  DO 900 KTIME = 1,NBRTIM
  DO 900 K = 1,19
    IF (K.EQ.1) FREQ = 260.0
    IF (K.EQ.2) FREQ = 395.0
    IF (K.EQ.3) FREQ = 535.0
    IF (K.EQ.4) FREQ = 683.0
    IF (K.EQ.5) FREQ = 841.0
    IF (K.EQ.6) FREQ = 1011.0
    IF (K.EQ.7) FREQ = 1198.0
    IF (K.EQ.8) FREQ = 1405.0
    IF (K.EQ.9) FREQ = 1635.0
    IF (K.EQ.10) FREQ = 1892.0
    IF (K.EQ.11) FREQ = 2179.0
    IF (K.EQ.12) FREQ = 2505.0
    IF (K.EQ.13) FREQ = 2885.0
    IF (K.EQ.14) FREQ = 3326.0
    IF (K.EQ.15) FREQ = 3855.0
    IF (K.EQ.16) FREQ = 4484.0
    IF (K.EQ.17) FREQ = 5263.0
    IF (K.EQ.18) FREQ = 6277.0
    IF (K.EQ.19) FREQ = 7626.0
CCC
    B(KTIME,K) = 0.0
    X0 = FREQ / 4.0
    X1 = FREQ / 2.0
    X2 = FREQ
    X3 = FREQ * 2.0
    X4 = FREQ * 4.0
CCC
    DO 900 J = 1,64
      X = (J * 78.125)
      FOFX = .244 + ((X-X0)*(1.6888)/FREQ)
      1 - ((X-X0)*(X-X1)*(1.363)/(FREQ**2))
      2 + ((X-X0)*(X-X1)*(X-X2)*(.6265)/(FREQ**3))
      3 - ((X-X0)*(X-X1)*(X-X2)*(X-X3)*(.14467)/(FREQ**4))
CCC
      IF (FOFX.GT.0.0) B(KTIME,K) = B(KTIME,K) + A(KTIME,J)*FOFX
CCC
  900 CONTINUE
  RETURN
  END

```

Explanation of Equation FOFX = . . .

The frequency response for each of Threshold Technology's nineteen frequencies was the following function:



This led to the following table of divided differences:

x	f(x)	$f(x_n, x_{n+1})$	$f(x_n, x_{n+1}, x_{n+2})$	$f(x_n \dots x_{n+3})$	$f(x_n \dots x_{n+4})$
F/4	.24444	1.6888/F	-1.363/F ²	.6265/F ³	-.144666/F ⁴
F/2	.6666	.666668/F	-.266667/F ²	.084/F ³	
F	1.0	-.266667/F	-.0277778/F ²		
2F	.7333	-.183383/F			
4F	.36666				

Thus, Newton's equation for f(x) becomes the equation FOFX used in subroutine NEWFREQ.

$$\text{(Newton's formula: } f(x) = f[x_0] + (x-x_0)f[x_0x_1] + (x-x_0)(x-x_1)f[x_0x_1x_2] \dots$$

```

PROGRAM GETDATA (INPUT,OUTPUT,TAPE9,TAPE1)
REWIND 9
REWIND 1
CALL BYPASS (45)
CALL IO (23,30,1,5,3)
CALL BYPASS (41)
CALL IO (25,30,1,5,3)
CALL BYPASS (40)
CALL IO (26,30,1,5,3)
CALL BYPASS (62)
CALL IO (28,30,2,5,3)
CALL BYPASS (42)
CALL IO (27,30,2,5,3)
CALL BYPASS (39)
CALL IO (30,30,2,5,3)
CALL BYPASS (42)
CALL IO (29,30,2,5,3)
CALL BYPASS (56)
CALL IO (38,40,3,5,3)
CALL BYPASS (36)
CALL IO (36,40,3,5,3)
CALL BYPASS (38)
CALL IO (43,40,3,5,3)
CALL BYPASS (39)
CALL IO (37,40,3,5,3)
CALL BYPASS (332)
CCCCCCCCCCCC
CALL IO (45,40,1,1,2)
CALL BYPASS (81)
CALL IO (32,30,2,1,2)
CALL BYPASS (74)
CALL IO (46,40,3,1,2)
CALL BYPASS (179)
CALL IO (44,40,1,1,2)
CALL BYPASS (78)
CALL IO (29,30,2,1,2)
CALL BYPASS (85)
CALL IO (44,40,3,1,2)
CALL BYPASS (134)
CALL IO (46,40,1,1,2)
CALL BYPASS (80)
CALL IO (32,30,2,1,2)
CALL BYPASS (88)
CALL IO (39,40,3,1,2)
CALL BYPASS (135)
CALL IO (59,40,1,1,2)
CALL BYPASS (64)
CALL IO (34,30,2,1,2)
CALL BYPASS (73)
CALL IO (40,40,3,1,2)
CALL BYPASS (430)

```

CCCCCCCCCCC

CALL IO (73,60,1,2,2)
CALL BYPASS (37)
CALL IO (39,40,2,2,2)
CALL BYPASS (50)
CALL IO (44,40,3,2,2)
CALL BYPASS (187)
CALL IO (60,60,1,2,2)
CALL BYPASS (44)
CALL IO (35,40,2,2,2)
CALL BYPASS (48)
CALL IO (52,40,3,2,2)
CALL BYPASS (126)
CALL IO (48,60,1,2,2)
CALL BYPASS (56)
CALL IO (30,40,2,2,2)
CALL BYPASS (64)
CALL IO (38,40,3,2,2)
CALL BYPASS (134)
CALL IO (52,60,1,2,2)
CALL BYPASS (53)
CALL IO (32,40,2,2,2)
CALL BYPASS (62)
CALL IO (39,40,3,2,2)
CALL BYPASS (500)

CCCCCCCCCCC

CALL IO (28,30,2,4,2)
CALL BYPASS (49)
CALL IO (50,40,3,4,2)
CALL BYPASS (118)
CALL IO (51,40,1,4,2)
CALL BYPASS (38)
CALL IO (30,30,2,4,2)
CALL BYPASS (47)
CALL IO (37,40,3,4,2)
CALL BYPASS (133)
CALL IO (41,40,1,4,2)
CALL BYPASS (15)
CALL IO (33,30,2,4,2)
CALL BYPASS (37)
CALL IO (39,40,3,4,2)
CALL BYPASS (413)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CALL IO (40,35,4,5,3)
CALL BYPASS (33)
CALL IO (35,35,4,5,3)
CALL BYPASS (30)
CALL IO (32,35,4,5,3)
CALL BYPASS (34)
CALL IO (35,35,4,5,3)
CALL BYPASS (54)
CALL IO (20,20,5,5,3)
CALL BYPASS (45)
CALL IO (21,20,5,5,3)
CALL BYPASS (45)
CALL IO (17,20,5,5,3)
CALL BYPASS (43)
CALL IO (18,20,5,5,3)
CALL BYPASS (64)
CALL IO (18,20,6,5,3)
CALL BYPASS (49)
CALL IO (17,20,6,5,3)
CALL BYPASS (45)
CALL IO (18,20,6,5,3)
CALL BYPASS (45)
CALL IO (18,20,6,5,3)
CALL BYPASS (69)
CALL IO (36,35,7,5,3)
CALL BYPASS (33)
CALL IO (37,35,7,5,3)
CALL BYPASS (34)
CALL IO (34,35,7,5,3)
CALL BYPASS (31)
CALL IO (30,35,7,5,3)
CALL BYPASS (443)

CCCCCCCCCCCCCCCC

CALL IO (32,30,4,1,1)
CALL BYPASS (112)
CALL IO (33,30,4,1,1)
CALL BYPASS (65)
CALL IO (30,30,4,1,1)
CALL BYPASS (170)
CALL IO (20,20,5,1,1)
CALL BYPASS (76)
CALL IO (20,20,5,1,1)
CALL BYPASS (73)
CALL IO (20,20,5,1,1)
CALL BYPASS (181)
CALL IO (16,20,6,1,1)
CALL BYPASS (86)
CALL IO (17,20,6,1,1)
CALL BYPASS (91)
CALL IO (16,20,6,1,1)
CALL BYPASS (191)
CALL IO (31,30,7,1,1)
CALL BYPASS (65)
CALL IO (28,30,7,1,1)
CALL BYPASS (72)
CALL IO (28,30,7,1,1)
CALL BYPASS (470)

CCCCCCCCCCCC

CALL IO (27,30,4,2,1)
CALL BYPASS (68)
CALL IO (28,30,4,2,1)
CALL BYPASS (70)
CALL IO (26,30,4,2,1)
CALL BYPASS (73)
CALL IO (29,30,4,2,1)
CALL BYPASS (143)
CALL IO (21,20,5,2,1)
CALL BYPASS (51)
CALL IO (23,20,5,2,1)
CALL BYPASS (61)
CALL IO (18,20,5,2,1)
CALL BYPASS (70)
CALL IO (18,20,6,2,1)
CALL BYPASS (57)
CALL IO (15,20,6,2,1)
CALL BYPASS (58)
CALL IO (20,20,6,2,1)
CALL BYPASS (189)
CALL IO (24,25,7,2,1)
CALL BYPASS (55)
CALL IO (28,25,7,2,1)
CALL BYPASS (51)
CALL IO (27,25,7,2,1)
CALL BYPASS (437)
CALL IO (17,20,5,3,1)
CALL BYPASS (43)
CALL IO (17,20,6,3,1)
CALL BYPASS (35)
CALL IO (25,25,7,3,1)
CALL BYPASS (173)
CALL IO (20,20,5,3,1)
CALL BYPASS (45)
CALL IO (17,20,6,3,1)
CALL BYPASS (99)
CALL IO (24,25,7,3,1)
STOP
END

C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C

```

SUBROUTINE IO (NBREC,IWRDSZ,IWORD,IGRAV,ISPKER)
  DIMENSION ARRAY (100,64)
  WRITE (1,*) IWORD,IGRAV,ISPKER,IWRDSZ
  DO 020 KOUNT = 1,NBREC
    READ (9) (ARRAY(KOUNT,I),I=1,64)
020 CONTINUE
  CALL MAKEPF (ARRAY,NBREC,IWRDSZ)
  RETURN
END

C
C
SUBROUTINE BYPASS (NBREC)
  DIMENSION ARRAY (64)
  DO 010 KOUNT = 1,NBREC
    READ (9) (ARRAY(I),I=1,64)
010 CONTINUE
  RETURN
END

C
C
SUBROUTINE MAKEPF (B,NBREC,IWRDSZ)
C THIS SUBROUTINE MAKEPF WILL CONVERT ARRAY B
C (WHICH CONTAINS THE ORIGINAL UNSCALED UNMODIFIED
C DATA) TO A STANDARD SIZE (IWRDSZ) AND THEN WRITE
C THIS WORD BLOCK TO A PERMFILE FOR INPUT TO SPSS.
C (EACH WORD AND GRAVITY WILL HAVE A STANDARD IWRDSZ;
C ITOTSZ WILL BE THE MAX IWRDSZ AND WILL SERVE AS A STANDARD
C TEMPLATE SIZE (ZERO FILLER) FOR SPSS INPUT. )
C
  DIMENSION B(100,64), C(100,64)
C
  ZERO = 0.0
  ITOTSZ = 60
  IF (NBREC.LT.IWRDSZ) CALL ADDTO (B,NBREC,IWRDSZ,C)
  IF (NBREC.GT.IWRDSZ) CALL CUT (B,NBREC,IWRDSZ,C)
  IF (NBREC.EQ.IWRDSZ) CALL BTOC (B,NBREC,C)
  DO 100 I = 1, IWRDSZ
    DO 100 J = 1,64
      WRITE (1,*) C(I,J)
100 CONTINUE
  IF (IWRDSZ.GE.ITOTSZ) GO TO 300
  IFILL = IWRDSZ + 1
  DO 200 I = IFILL, ITOTSZ
    DO 200 J = 1,64
      WRITE (1,*) ZERO
200 CONTINUE
300 RETURN
END

C
C

```

```

      SUBROUTINE BTOC (B,NBREC,C)
C   THIS SUBROUTINE COPIES ARRAY B TO ARRAY C WITH NO
C   CHANGES
C
      DIMENSION B(100,64), C(100,64)
      DO 100 I = 1,NBREC
        DO 90 J = 1,64
          C(I,J) = B(I,J)
        90 CONTINUE
      100 CONTINUE
      RETURN
      END

C
C
      SUBROUTINE ADDTO (B,NBREC,IWRDSZ,C)
C   THIS ROUTINE PADS ARRAY C TO IWRDSZ FOR WRITING
C   TO A PERMFILE FOR LATER INPUT TO SPSS.
C
      DIMENSION B(100,64), C(100,64)
C
      NBRADD = IWRDSZ - NBREC
      NTH = (NBREC / (NBRADD + 1)) + 1
      NHALF = NTH / 2
      IB = 1
      IC = 1
      IF ((NTH*NBRADD).LT.NBREC) GO TO 10
      DO 050 J = 1,NHALF
        DO 45 K = 1,64
          C(IC,K) = B(IB,K)
        45 CONTINUE
        IB = IB + 1
        IC = IC + 1
      50 CONTINUE
      10 CONTINUE
        DO 100 J = 1,NTH
          DO 90 K = 1,64
            C(IC,K) = B(IB,K)
          90 CONTINUE
          IB = IB + 1
          IC = IC + 1
        IF (IB.GT.NBREC) GO TO 990
      100 CONTINUE

```

```

C
C ADD AN NTH LINE TO C VIA AN AVERAGE OF B(IB) AND B(IB+1)
C
C THERE WILL BE A LOGICAL ERROR PROBLEM BELOW IF NBRADD
C IS .GE. NBREC. THE PROBLEM IS IGNORED FOR NOW
C
      DO 105 K = 1,64
        C(IC,K) = ((B(IB,K) + B(IB-1,K)) / 2)
105 CONTINUE
      IC = IC + 1
      GO TO 10
990 CONTINUE
      IF (IC.GT.IWRDSZ) GO TO 999
      DO 110 K = 1,64
        C (IC,K) = B(NBREC,K)
110 CONTINUE
999 CONTINUE
      RETURN
      END

C
C
      SUBROUTINE CUT (B,NBREC,IWRDSZ,C)
C THIS SUBROUTINE CUTS DOWN ARRAY C TO IWRSZ
C FOR WRITING TO PERMFILE FOR INPUT TO SPSS.
C
      DIMENSION B(100,64), C(100,64)
      NBRCUT = NBREC - IWRDSZ
      NTH = (NBREC / (NBRCUT + 1))
      IB = 1
      IC = 1
10 CONTINUE
      DO 100 J = 1,NTH
        DO 90 K = 1,64
          C(IC,K) = B(IB,K)
90 CONTINUE
        IB = IB + 1
        IC = IC + 1
        IF (IC.GT.IWRDSZ) GO TO 999
100 CONTINUE

C
C CUT THE NTH LINE OF IB
C
      IB = IB + 1
      GO TO 10
999 CONTINUE
      RETURN
      END

```

```

PROGRAM TEST(INPUT,OUTPUT,TAPE9,TAPE5=INPUT,
+           TAPE6=OUTPUT)
      DIMENSION A(150,68), FMAT(80), CMNT(80)
10  FORMAT(80A1)
CCCC15 FORMAT(1H1,80A1)
20  FORMAT(1H ,5(A6,F7.2))
      PRINT 30
30  FORMAT(1H , "INPUT RTSMAG,IWDGAP,INOPT,IPROPT",/)
      READ(5,*) RTSMAG,IWDGAP,INOPT,IPROPT
      PRINT 35
35  FORMAT(1H , "INPUT A FORMAT OF UP TO 80 CHARACTERS",/)
      READ(5,10) FMAT
      PRINT 40
40  FORMAT(1H , "INPUT A TITLE OF UP TO 80 CHARACTERS",/)
      READ(5,10) CMNT
      IFRST=1
      IRLAST=0
      ICFRST=1
      ICLAST=64
      IRA=150
      ICA=68
      ICTS=66
      ICSSQ=67
      ICNORM=68
      INTSRD=0
      INFILE = 9
      IOFILE = 6
      WRITE(6,101)
12  CONTINUE
      CALL GETTSS(A,IRA,ICA,IFRST,IRLAST,ICFRST,ICLAST,ICTS,ICSSQ,
$           INFILE,FMAT,RTSMAG,IWDGAP,INTSRD)
      IF (IRLAST.LE.IFRST) GOTO 100
CCCCC WRITE(6,15) (CMNT(I),I=1,80)
CCCCC WRITE(6,*) "IRFRST",IRFRST,"IRLAST",IRLAST,"ICFRST",ICFRST,
CCCCC$ "ICLAST",ICLAST
CCCCC WRITE(6,*) "RTSMAG",RTSMAG,"IWDGAP",IWDGAP,"INTSRD",INTSRD,
CCCCC$ "INOPT ",INOPT,"IPROPT",IPROPT
      CALL NORM(A,IRA,ICA,IFRST,IRLAST,ICFRST,ICLAST,ICTS,
$           ICSSQ,ICNORM,INOPT)
      WRITE(6,*) "SPECTROGRAM"
      CALL SPECT(A,IRA,ICA,IFRST,IRLAST,ICFRST,ICLAST,ICTS,ICSSQ,
$           ICNORM,IOFILE,INOPT,IPROPT)
      GOTO 12
100  CONTINUE
      WRITE(6,102)
101  FORMAT(1HT,"T SETS PRINTER TO 8 LINES PER INCH")
102  FORMAT(1HS,"S SETS PRINTER TO 6 LINES PER INCH")
      STOP
      END

```

```

      FUNCTION RMAXEL(A,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C**** FUNCTION RMAXEL(A,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C****
C****      THIS FUNCTION RETURNS THE MAXIMUM ELEMENT CONTAINED IN
C****      THE DIMENSION ARRAY BETWEEN ROWS IRLAST AND ICLAST INCLUSIVE
C****      BETWEEN COLUMNS ICFRST AND ICLAST INCLUSIVE IN EACH ROW
C****
C**** INPUT
C****      ARRAY IS ANY DIMENSION
C****      IRA IS THE TRUE ROW SIZE OF A
C****      ICA IS AN APPROPRIATE COLUMN SIZE FOR A,USUALLY EQUAL TO ICLAST
C****      IRFRST IS THE STARTIN ROW FOR THE MAX ELEMENT SEARCH
C****      IRLAST IS THE ENDING ROW FOR THE MAX ELEMENT SEARCH
C****      ICFRST IS THE STARTING COLUMN IN EACH ROW FOR THE SEARCH
C****      ICLAST IS THE ENDING OLUMN IN EACH ROW FOR THE SEARCH
C****
      DIMENSION A(IRA,ICA)
      RMAXEL = A(IRFRST,1)
      DO 200 I=IRFRST,IRLAST
        DO 100 J=ICFRST,ICLAST
          IF (A(I,J). GT .RMAXEL) RMAXEL = A(I,J)
100      CONTINUE
200 CONTINUE
      RETURN
      END
      FUNCTION SUMEL(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C****
C**** FUNCTION SUMEL(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C****      THIS FUNCTION RETURNS THE SUMS OF COLUMNS ICFRST THROUGH ICLAST
C****      OVER ROWS IRFRST THROUGH IRLAST INCLUSIVE IN DIMENSION ARRAY
C****
C**** INPUT
C****      ARRAY IS ANY DIMENSION
C****      IAR IS THE TRUE ROW SIZE OF ARRAY
C****      IAC IS USED TO SPECIFY THE COLUMN DIMENSION OF ARRAY AND SHOULD
C****      SET TO AN APPROPRIATE VALUE
C****      IRFRST IS THE STARTING ROW FOR THE SUMMATION
C****      IRLAST IS THE ENDING ROW FOR THE SUMMATION
C****      ICFRST IS THE STARTING COLUMN IN EACH ROW FOR THE SUMMATION
C****      ICLAST IS THE ENDING COLUMN IN EACH ROW FOR THE SUMMATION
      DIMENSION ARRAY(IRA,ICA)
      SUMEL = 0.0
      DO 200 I=IRFRST,IRLAST
        DO 100 J=ICFRST,ICLAST
          SUMEL = SUMEL + ARRAY(I,J)
100      CONTINUE
200 CONTINUE
      RETURN
      END

```



```

        FUNCTION SUMSQR(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C****
C**** FUNCTION SUMSQR(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C****
C****     THIS FUNCTION COMPUTES THE SUM OF SQUARES OF THE ELEMENTS
C****     BETWEEN COLUMNS ICFRST AND ICLAST OF ROWS IRFRST THROUGH
C****     IRLAST INCLUSIVE IN THE DIMENSION ARRAY
C****
C**** INPUT
C****     ARRAY IS ANY DIMENSION
C****     IRA IS THE TRUE ROW DIMENSION OF ARRAY
C****     ICA IS THE APPROPRIATE COLUMN DIMENSION FOR ARRAY
C****     IRFRST IS THE STARTING ROW FOR THE CALCULATION
C****     IRLAST IS THE ENDING ROW FOR THE CALCULATION
C****     ICFRST IS THE STARTING COLUMN IN EACH ROW FOR THE CALCULATION
C****     ICLAST IS THE ENDING COLUMN IN EACH ROW FOR THE CALCULATION
C****
        DIMENSION ARRAY(IRA,ICA)
        SUMSQR = 0.0
        DO 200 I=IRFRST,IRLAST
            DO 100 J=ICFRST,ICLAST
                SUMSQR = SUMSQR + ARRAY(I,J)*ARRAY(I,J)
100     CONTINUE
200 CONTINUE
        RETURN
        END

```

```

SUBROUTINE GETTSS(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST,
$    ICTS,ICSSQ,INFILE,FMAT,RTSMAG,IWDGAP,INTSRD)
C****
C**** SUBROUTINE GETTSS(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST
C****    ICTS,ICSSQ,INFILE,FMAT,RTSMAG,IWDGAP,INTSRD)
C****
C**** THIS SUBROUTINE STARTS AT THE CURRENT RECORD OF THE INPUT FILE
C**** INFILE AND READS A WORD INTO THE DIMENSION ARRAY FROM INFILE.
C**** WORD START LOCATION IS DETERMINED BY READING UNTIL SOME
C**** INPUT TIME SLICE HAS ELEMENTS WITH SUM OF SQUARES > RTSMAG.
C**** FROM THIS POINT THE WORD IS CONSIDERED TO CONTINUE UNTIL
C**** IWDGAP CONSECUTIVE TIME SLICES EACH HAVE ELEMENTS WITH SUM
C**** OF SQUARES <= RTSMAG. THE WORD END IS THEN SET AT TWENTY
C**** TIME SLICES BEFORE THE LAST TIME SLICE READ. STORAGE IN
C**** ARRAY STARTS AT ROW IRFRST. THE TIME SLICE NUMBER OF EACH
C**** SLICE WITHIN INFILE (BASED ON INTSRD) AND THE SUM OF SQUARES
C**** OF EACH TIME SLICE (THE ENERGY OF THE TIME SLICE) ARE ALSO
C**** RETURNED AS DESCRIBED BELOW.
C****
C**** INPUT
C**** ARRAY IS ANY DIMENSION
C**** IRA IS THE TRUE ROW DIMENSION OF ARRAY
C**** ICA IS A COLUMN DIMENSION FOR ARRAY SET APPROPRIATELY BUT MUST
C**** BE EQUAL TO (ICLAST-ICFRST)+2 AS A MINIMUM (SEE OTHER INPUT
C**** PARAMETER DESCRIPTIONS BELOW)
C**** IRFRST IS THE ROW IN ARRAY AT WHICH STORAGE IS TO BEGIN
C**** IRLAST IS USED FOR OUTPUT
C**** ICFRST IS THE COLUMN IN EACH ROW OF ARRAY AT WHICH STORAGE
C**** OF COEFFICIENTS IS TO BEGIN
C**** ICLAST IS THE COLUMN IN EACH ROW OF ARRAY AT WHICH STORAGE
C**** OF COEFFICIENTS IS TO END *****NOTE THAT NUMBER OF
C**** COEFFICIENTS = (ICLAST-ICFRST)+1
C**** ICTS IS THE COLUMN NUMBER IN EACH ROW OF ARRAY IN WHICH
C**** THE TIME SLICE NUMBER OF THE TIME SLICE IS TO BE STORED
C**** ICSSQ IS THE COLUMN NUMBER IN EACH ROW OF ARRAY IN WHICH
C**** THE SUM OF SQUARES OF THE ELEMENTS OF THE TIME SLICE IS
C**** TO BE STORED
C**** INFILE IS THE TAPE NUMBER OF THE FILE FROM WHICH READING IS
C**** TO TAKE PLACE
C**** FMAT IS AN ARRAY OF 80 CHARACTERS,ONE CHARACTER PER WORD
C**** WITH WHICH TO READ THE INPUT DATA
C**** RTSMAG IS THE REAL MAGNITUDE CUTOFF FOR SUM OF SQUARES OF
C**** ELEMENTS ON WHICH TIME SLICE INCLUSION IN A WORD IS BASED
C**** IWDGAP IS THE ESTIMATE OF THE MINIMUM GAP (IN NUMBER OF
C**** TIME SLICES) BETWEEN WORDS
C**** INTSRD IS THE NUMBER OF TIME SLICES READ FROM INFILE BEFORE
C**** THIS CALL TO GETTSS

```

```

C**** OUTPUT
C****   ARRAY CONTAINS A WORD READ FROM INFILE. STORAGE BEGINS IN ROW
C****   IRFRST AND ENDS IN ROW IRLAST. COEFFICIENTS IN EACH ROW
C****   ARE STORED FROM COLUMN ICFRST THROUGH COLUMN ICLAST. THE
C****   TIME SLICE NUMBER OF EACH TIME SLICE IS IN COLUMN ICTS OF
C****   EACH ROW AND THE SUM OF SQUARES OF THE COEFFICIENTS IS IN
C****   COLUMN ICSSQ OF EACH ROW.
C****   IRLAST IS THE ROW IN WHICH STORAGE OF THE WORD ENDS
C****   ALL OTHER PARAMETERS ARE UNCHANGED
C**** SPECIAL NOTES
C****   THE EXTERNAL FUNCTION SUMSQR IS USED FOR FINDING THE
C****   SUM OF SQUARES OF COEFFICIENTS IN EACH TIME SLICE
C****   THE SIZE OF ARRAY IN ROWS SHOULD BE AT LEAST THE MAXIMUM
C****   NUMBER OF TIME SLICES EXPECTED IN A WORD PLUS IWDGAP
C****   PLUS IRFRST MINUS 1
C****   EACH TIME SLICE IS ASSUMED TO CONTAIN ICLAST-ICFRST+1
C****   COEFFICIENTS
C****   IF THE NUMBER OF TIME SLICES IN A WORD WOULD CAUSE
C****   A SUBSCRIPT TO ARRAY LARGER THAN IRA (ARRAY'S TRUE ROW SIZE)
C****   THE ROUTINE WILL EXIT AND ARRAY WILL CONTAIN ALL TIME
C****   SLICES READ SO FAR. THE CONDITION IS INDICATED BY
C****   HAVING IRLAST=IRA UPON SUBROUTINE EXIT
C****   IF AN EOF ON INFILE IS REACHED BEFORE A WORD IS COMPLETED
C****   (IN THE SENSE OF IWDGAP LIMITS DESCRIBED) THE WORD IS
C****   CONSIDERED TO HAVE ENDED AT THE END OF THE FILE AND
C****   IRLAST IS SET APPROPRIATELY
C****
C****   DIMENSION ARRAY(IRA,ICA), FMAT(80)
C****   THIS SECTION READS FROM CURRENT RECORD TO BEGINNING OF A WORD
C****   AS DETERMINED BY SETTING OF RTSMAG
C****
C****   I=IRFRST
100 READ(INFILE) (ARRAY(I,J),J=ICFRST,ICLAST)
C**
C**   CALL SCALE(ARRAY,IRA,ICA,I)
C**
C**   THIS CALL TO SCALE SHOULD ONLY BE MADE
C**   TO EQUALIZE 64 CHANNELS OF INPUT
C**   (EQUALIZE HIGH AND LOW FREQUENCIES)
C**   BEFORE THE SPECTROGRAPH IS NORMALIZED
C**   AND PRINTED (LEVINE - 23 JUN 80)
C**
C**   IF (EOF(INFILE).EQ.1) GOTO 1000
C**   INTSRD = INTSRD + 1
C**   ARRAY(I,ICSSQ) = SUMSQR(ARRAY,IRA,ICA,I,I,ICFRST,ICLAST)
C**   IF (ARRAY(I,ICSSQ).LT.RTSMAG) GOTO 100

```

```

C***
C*** FALL THROUGH TO THIS SECTION TO READ A WORD
C*** WHERE READING PROCEEDS TO WORD END PLUS IWDGAP
C*** TIMESLICES WHOSE MAGNITUDE IS LESS THAN RTSMAG
C***
C*** FIRST SAVE TIME SLICE NUMBER OF LAST TIME SLICE READ ABOVE
C*** SINCE IT IS THE FIRST TIME SLICE IN THE WORD
C***
      ARRAY(I,ICTS)=FLOAT(INTSRD)
490 NTSSKP=0
500 I=I+1
      IF (I.GT.IRA) GOTO 1100
      READ(INFILE) (ARRAY(I,J),J=ICFRST,ICLAST)
C**
      CALL SCALE(ARRAY,IRA,ICA,I)
C**
      IF (EOF(INFILE).EQ.1) GOTO 1000
      INTSRD=INTSRD + 1
      ARRAY(I,ICTS) = FLOAT(INTSRD)
      ARRAY(I,ICSSQ) = SUMSQR(ARRAY,IRA,ICA,I,I,ICFRST,ICLAST)
      NTSSKP = NTSSKP + 1
      IF (ARRAY(I,ICSSQ).GT.RTSMAG) GOTO 490
      IF (NTSSKP.LT.IWDGAP) GOTO 500
      IRLAST=I-IWDGAP
      RETURN
C***
C*** GET TO 1000 IF EOF REACHED ON INFILE
1000 CONTINUE
      IRLAST=I-1
      RETURN
C***
C*** GET TO 1100 IF ROW SIZE OF A WAS ABOUT TO BE EXCEEDED
1100 CONTINUE
      IRLAST=IRA
      RETURN
      END

```

```

      SUBROUTINE NORM(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST,
$      ICTS,ICSSQ,ICNORM,IOPT)
C****
C**** SUBROUTINE NORM(ARRAY,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST,
C****      ICTS,ICSSQ,ICNORM,IOPT)
C****
C**** THIS SUBROUTINE NORMALIZES A SECTION OF THE DIMENSION ARRAY.
C**** THE SECTION NORMALIZED IS FROM ROW IRFRST THROUGH ROW IRLAST
C**** OVER COLUMNS ICFRST THROUGH ICLAST. THE ROUTINE EXPECTS TO
C**** FIND THE SUM OF SQUARES OF COLUMNS ICFRST THROUGH ICLAST IN
C**** COLUMN ICSSQ OF EACH ROW. COLUMN ICTS OF EACH ROW IS ASSUMED
C**** TO BE AN IDENTIFYING NUMBER BUT IS NOT USED IN THE CURRENT
C**** VERSION OF THE SUBROUTINE. THE NORMALIZATION FACTOR USED TO
C**** NORMALIZE A PARTICULAR ROW OF A IS RETURNED IN COLUMN ICNORM
C**** OF EACH ROW. THE NORMALIZATION PERFORMED DEPENDS ON THE SETTING OF
C**** IOPT AS DISCUSSED UNDER INPUT DESCRIPTION
C****
C**** INPUT
C****      ARRAY IS ANY DIMENSION
C****      IRA IS THE TRUE ROW DIMENSION OF ARRAY
C****      ICA IS THE MAXIMUM COLUMN NUMBER USED IN ARRAY AND MUST BE
C****      EQUAL TO AT LEAST (ICLAST-ICFRST)+4
C****      IRFRST IS THE ROW IN ARRAY AT WHICH NORMALIZATION IS TO START
C****      IRLAST IS THE ROW IN ARRAY AT WHICH NORMALIZATION IS TO END
C****      ICFRST IS THE STARTING COLUMN IN EACH ROW AT WHICH NORMALIZA-
C****      TION IS TO START
C****      ICLAST IS THE ENDING COLUMN IN EACH ROW AT WHICH NORMALIZA-
C****      TION IS TO END
C****      ICTS IS A COLUMN NUMBER WITHIN EACH ROW ASSUMED TO CONTAIN
C****      AN IDENTIFYING NUMBER, NOT USED IN CURRENT VERSION OF
C****      SUBROUTINE
C****      ICSSQ IS THE COLUMN NUMBER WITHIN EACH ROW WHICH SHOULD
C****      CONTAIN THE SUM OF SQUARES OF THE COLUMNS ICFRST THRU ICLAST
C****      ICNORM IS THE COLUMN NUMBER WITHIN EACH ROW IN WHICH THE
C****      NORMALIZATION FACTOR USED FOR THAT ROW IS TO BE STORED
C****      IOPT IS THE NORMALIZATION OPTION AS FOLLOWS
C****          2 NORMALIZES EACH INDIVIDUAL ROW WITHIN ITSELF
C****          3 NORMALIZES OVER THE ENTIRE RANGE OF ROWS IRFRST
C****          THROUGH IRLAST
C****
C**** OUTPUT
C****      ARRAY - EACH ROW OF ARRAY CONTAINS COLUMNS AS FOLLOWS:
C****          NCFRST THRU NCLAST CONTAIN NORMALIZED TERMS (ORIGINAL
C****          COLUMNS NORMALIZED ACCORDING TO IOPT )
C****          ICTS IS UNCHANGED, ASSUMED TO BE A TIME SLICE NUMBER
C****          ICSSQ CONTAINS THE TOTAL ENERGY IN THE ROW = SUM OF
C****          RETURNED ELEMENTS NCFRST THRU NCLAST
C****          ICNORM CONTAINS THE NORMALIZATION FACTOR USED TO
C****          NORMALIZE THAT ROW

```

```

C****
C**** SPECIAL NOTES
C**** VARIABLE COLUMN DIMENSION ICA MUST BE EQUAL TO A MINIMUM OF
C**** ((ICLAST-ICFRST)+4) OR THE MAXIMUM OF(ICLAST,ICFRST,ICTS,
C**** ICSSQ,ICNORM), WHICHEVER CAUSES THE LARGEST DIMENSION
C**** THE COLUMN NUMBERS OF ICTS,ICSSQ, AND ICNORM SHOULD ALL BE
C**** < ICFRST AND/OR > ICLAST (I.E. SHOULD NOT LIE BETWEEN
C**** ICFRST AND ICLAST INCLUSIVE)
C**** IF THIS SUBROUTINE IS EXECUTED WITH IOPT SET TO A VALUE OTHER
C**** THAN 2 OR 3 IT EXITS WITHOUT AFFECTING ARRAY
C****
      DIMENSION ARRAY(IRA,ICA)
      IF (IOPT.GT.3) RETURN
      GOTO (9999,2000,3000),IOPT
C***   ABOVE CODE BRANCHES TO APPROPRIATE OPTION,EXITS ON BAD OPTION
C***
2000 CONTINUE
C***   IOPT=2,NORMALIZE EACH ROW WITHIN ITSELF
      DO 2100 I=IRFRST,IRLAST
        ARRAY(I,ICNORM)=SQRT(ARRAY(I,ICSSQ))
        DO 2050 J=ICFRST,ICLAST
          ARRAY(I,J) = ARRAY(I,J) / ARRAY(I,ICNORM)
2050 CONTINUE
2100 CONTINUE
      RETURN
C***
3000 CONTINUE
C***   IOPT = 3 NORMALIZE OVER ENTIRE WORD
      SUMOSUM = 0.0
      DO 3100 I=IRFRST,IRLAST
        SUMOSUM = SUMOSUM + ARRAY(I,ICSSQ)
3100 CONTINUE
      RNORM = SQRT(SUMOSUM)
      DO 3200 I=IRFRST,IRLAST
        DO 3150 J=ICFRST,ICLAST
          ARRAY(I,J) = ARRAY(I,J) / RNORM
3150 CONTINUE
        ARRAY(I,ICNORM) = RNORM
3200 CONTINUE
      RETURN
C***
C*** NEXT IS COMMON RETURN/EXIT FOR BAD OPTIONS WHEN IOPT
C**** AN INVALID OPTION NUMBER
9999 RETURN
      END

```

```

      SUBROUTINE SPECT(A,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST,
$      ICTS,ICSSQ,ICNORM,IOFILE,INOPT,IPROPT)
C**** SUBROUTINE SPECT(A,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST,
C****   ICTS,ICSSQ,ICNORM,IOFILE,INOPT,IPROPT)
      DIMENSION A(IRA,ICA), SPECLN(66)
      20 FORMAT(1H ,F5.0,1X,F10.3,1X,66A1)
      30 FORMAT(1H+,17X,66A1)
      40 FORMAT(1H+,35X,16F6.3)
C***  ***INITIALIZE ARRAY SPECLN,WHICH HOLDS THE SPECTROGRAM LINES
      NCOEFF = ICLAST - ICFRST + 1
      ILNGTH = ICLAST - ICFRST + 3
      DO 50 I=1,ILNGTH
        SPECLN(I) = " "
      50 CONTINUE
      SPECLN(1) = "$"
      SPECLN(ILNGTH) = "$"
C***  ***SET RNG OVER WHICH COEFFICIENTS RUN DEPENDING ON THE NORMALI-
C***  ***ZATION OPTION SPECIFIED, I.E. DEPENDING
C***  ***ON SETIN OF INOPT
      IF (INOPT.EQ.1) RNG = 100.0
      IF (INOPT.EQ.3) RNG=RMAXEL(A,IRA,ICA,IRFRST,IRLAST,ICFRST,ICLAST)
C***  ***NOTE NO SETTING FOR INOPT=3, SETTING IS SET FOR EACH
C***  ***INDIVIDUAL ROW AT PRINT TIME

```

```

C***
C*** ***DO LOOP THROUGH 1500 LOOPS THROUGH ROWS
C*** ***TO ACCOMPLISH CREATION OF SPECTROGRAM
DO 1500 J=IRFRST,IRLAST
C*** ***PROCESS FIRST LINE OF SPECTROGRAM
IF (INOPT.EQ.2) RNG= RMAXEL(A,IRA,ICA,J,J,ICFRST,ICLAST)
C*** ***SPECTROGRAM CREATION NEXT ,FIRST LINE OF TWO
L=1
DO 1200 K=ICFRST,ICLAST
L=L+1
SPECLN(L)="I"
IF (A(J,K).LE.(RNG*0.5)) SPECLN(L)="="
IF (A(J,K).LE.(RNG*0.4)) SPECLN(L)="*"
IF (A(J,K).LE.(RNG*0.3)) SPECLN(L)="+"
IF (A(J,K).LE.(RNG*0.2)) SPECLN(L)="-"
IF (A(J,K).LE.(RNG*0.1)) SPECLN(L)=" "
1200 CONTINUE
WRITE(IOFILE,20) A(J,ICTS),A(J,ICSSQ),(SPECLN(K),K=1,ILNGTH)
C*** ***PROCESS SECOND LINE OF SPECTROGRAM
L=1
DO 1400 K=ICFRST,ICLAST
L=L+1
SPECLN(L)="H"
IF (A(J,K).LE.(RNG*0.9)) SPECLN(L)="#"
IF (A(J,K).LE.(RNG*0.8)) SPECLN(L)="X"
IF (A(J,K).LE.(RNG*0.7)) SPECLN(L)="="
IF (A(J,K).LE.(RNG*0.6)) SPECLN(L)="-"
IF (A(J,K).LE.(RNG*0.4)) SPECLN(L)=" "
1400 CONTINUE
WRITE(IOFILE,30) (SPECLN(K),K=1,ILNGTH)
C*** ***NOW PRINT COEFFICIENTS AND/OR OTHER INFORMATION
IF ((IPROPT.EQ.2). AND .(NCOEFF.LE.16))
$ WRITE(IOFILE,40) (A(J,K),K=ICFRST,ICLAST)
1500 CONTINUE
RETURN
END
C**

```



```

SUBROUTINE SCALE(ARRAY,IIRA,IICA,II)
C**
C** "II" REPRESENTS THE ROW OF DATA JUST READ IN
C** WHICH IS TO BE EQUALIZED TO THE SAME SCALE
C** USED TO REDUCE 64 CHANNELS TO 16
C** (LEVINE 23 JUN 80 )
C**
      DIMENSION ARRAY(IIRA,IICA)
C**
      DO 10 K=7,12
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 2
10    CONTINUE
C**
      DO 20 K=13,20
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 4
20    CONTINUE
C**
      DO 30 K=21,25
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 5
30    CONTINUE
C**
      DO 40 K=26,31
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 6
40    CONTINUE
C**
      DO 50 K=32,40
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 9
50    CONTINUE
C**
      DO 60 K=41,50
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 10
60    CONTINUE
C**
      DO 70 K=51,64
        ARRAY(II,K) = (ARRAY(II,K) - 0) * 14
70    CONTINUE
      RETURN
      END

```

VITA

Nadine E. Levine was born on 26 October 1950 in Brooklyn, New York. She graduated from Jericho High School in 1968 and from Michigan State University in 1972, with a Bachelor of Science Degree in Mathematics. She entered the United States Air Force in October 1974, and after three months in OTS served from February 1975 through June 1979 as a programmer/analyst for MAC/AD at Scott AFB, Illinois. During this time she served as officer in charge of the team that automated MAC Airlift Planning. In June 1979 she was assigned to the Air Force Institute of Technology as a graduate student in Computer Systems.

Permanent Address: 651 N.W. 76th Terrace
Margate, Florida 33063

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/EE/80D-11	2. GOVT ACCESSION NO. AD-A100736	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) APPLICATION OF DISCRIMINANT ANALYSIS TECHNIQUES TO G-STRESS ANALYSIS IN COMPUTER VOICE DECODING		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) Nadine E. Levine		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 64
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 <i>Fredric C. Lynch</i> FREDRIC C. LYNCH, Major, USAF Director of Public Affairs 16 JUN 1981		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Speech Recognition, Gravity, G-Stress, Voice Decoding, Discriminant Analysis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Statistical Package for the Social Sciences (SPSS) Discriminant Analysis Routines were applied to speech data obtained at various gravity levels. The data were created on the Air Force Medical Research Lab's Centrifuge at Wright-Patterson Air Force Base. They were then digitally sampled and Fourier transformed to 64 frequency bands which were converted to two separate files; one the 16 frequencies used by Guyote and Sisson; one the 19 frequencies used by Threshold Technology, Inc. These files were		

then further processed: summing the energy across time slices, summing across frequency bands, and averaging across frequency bands. Statistical results showed that G-stress seemed to be moving energy across time slices, more than shifting across frequencies.

DATE
FILMED
-8